# CYBERLOGIC OPC SERVER

## Version 6.00 for Windows® XP/2000/NT/Server 2003



Document last revision date January 30, 2006

# TABLE OF CONTENTS

# INTRODUCTION

The Cyberlogic OPC Server has a modular structure that supports a variety of industrial devices and communication networks. The various communication subsystems, which we call driver agents, are plug-ins that you can easily add as required. As a result, the Server maintains a set of common features, but has the flexibility to allow additional features as required by the specific driver agent.

The Cyberlogic OPC Server has several unique features that provide top performance along with reliability and ease of use. It uses an advanced transaction optimizer to guarantee minimum load on the communication networks. The multithreaded design ensures minimum system loading and, at the same time, delivers unmatched data transfer speeds.

The Server supports multiple, priority-based, access paths for backup and redundant communications. In case of a communication path failure, the Server automatically switches over to the next available access path. The Server continues to monitor all failed access paths so that it can switch back to a higher priority path when one becomes available.

You can configure data tags for solicited or unsolicited data updates. An advanced unsolicited message filter allows the Server to filter out data messages from unwanted data sources, ensuring better data integrity and security. Other noteworthy features include direct data access, data write protection and a health watchdog.

A sophisticated, but intuitive, configuration editor allows easy configuration of the hierarchical tag database. With only a few mouse clicks, the Cyberlogic OPC Server can automatically configure its basic communication functions.

This document describes only the common features of the Cyberlogic OPC Server. For information related to a particular driver agent, refer to the help file specific for that Agent.

# Compatibility and Features

The Cyberlogic OPC Server is compatible with all local and remote OPC Data Access and Alarms & Events clients, including HMI, SCADA, ActiveX Controls and custom VB and C/C++ applications. It provides full compliance with the OPC Data Access 3.0, 2.05a and 1.0a specifications as well as Alarms and Events 1.1. It is also compliant with XML Data Access 1.0 and OPC Data Access Automation 2.02.

Some of the major features are:

- Modular structure allows support for a variety of industrial devices and communication networks
- Supports solicited and unsolicited communications
- Supports multiple priority-based access paths for backup and redundant communications
- Automatically switches to the next available access path when the current data source connection fails. Automatically switches back to a higher priority access path when one becomes available.
- Supports health-monitoring of access paths
- Sophisticated unsolicited message filter permits filtering out data messages from unwanted sources
- Flexible, multilevel, tag address space permits logical grouping of tags that is independent of the physical device data layout
- Supports on-line configuration changes
- Keeps track of recent edits and allows undo operations
- Powerful import/export feature permits quick data tag configuration by importing from Cyberlogic and other OPC server products
- High performance due to advanced multithreaded design
- Advanced transaction optimizer guarantees minimum loading of the communication networks
- DirectAccess to all registers in any device with minimal server configuration
- Write protection based upon individual data item, device or group of devices
- Supports a variety of data types at the data source
- Built-in data conversions
- Built-in data simulation
- Operates in the background as a service
- Integrated real-time Data Monitor

# What Should I Do Next?

This document describes only the common features of the Cyberlogic OPC Server. For information related to a particular driver agent, refer to the help file specific for that Agent.

For architectural and implementation details of the Cyberlogic OPC Server, read the Theory of Operation section. This section describes the implementation of various features of the Server, as well as troubleshooting aids.

You must configure the Cyberlogic OPC Server after installing it. You will find information on this topic in the Configuration section. This section contains a step-by-step tutorial for first-time users along with a detailed explanation of the configuration tools.

If you have already configured the Server, refer to the Validation & Troubleshooting section to verify that it operates as expected, and for problem solving hints.

This document is also provided in the PDF file format. You can use the Adobe® Reader program to view and print the PDF files.

# THEORY OF OPERATION

This section will familiarize you with the main features of the Cyberlogic OPC Server.

OPC (OLE for Process Control) is based upon Microsoft's COM/DCOM architecture and allows client applications access to plant floor data in a consistent manner. The OPC specification defines a set of industry specific COM interfaces through which applications can read and write data to a variety of industrial devices.

The Cyberlogic OPC Server provides full compliance with the OPC Data Access 3.0, 2.05a and 1.0a specifications as well as Alarms and Events 1.1. It is also compliant with XML Data Access 1.0 and OPC Data Access Automation 2.02.

## Main Server Features

The Cyberlogic OPC Server supports multiple priority-based access paths for redundant communications. Data items can be configured for both solicited and unsolicited data updates. The DirectAccess feature allows an OPC client to connect to a minimally-configured Server and still access any registers in the device by using the register address. A sophisticated transaction optimizer guarantees minimum loading of the communication networks. The Server's advanced multithreaded design ensures minimum system loading while delivering unmatched data transfer speeds.

The Cyberlogic OPC Server Configuration Editor allows easy configuration of the Server's hierarchical tag database. It supports advanced features such as the automatic detection of network connections and nodes, configuration import/export and a built-in Data Monitor.

The following sections describe these operational features of the Server.

## Network Connections and Network Nodes

Driver agents use various means for connecting to their devices or networks. In some cases a serial COM port serves that purpose. In other cases, a network card is used. The Cyberlogic OPC Server refers to all of these using the generic term, Network Connection.

For example, in the Cyberlogic DHX architecture, DHX devices represent Network Connections. In some cases a DHX device corresponds to a physical network card, such as a 1784-PKTX. In other cases, it is an abstract object, such as an Ethernet DHX device, that behaves like a network card.

The Server refers to each physical device on the network as a Network Node. A typical Network Node might be a PLC-5 on a Data Highway network. The Server accesses Network Nodes through their corresponding Network Connection. The Network Node configuration contains the communication parameters for the physical node device.

Conceptually, then, a Network Connection represents a single network with at least one Network Node. Even if the underlying communication port allows communications only to a single device, it is still viewed as a network with a single node.

For most driver agents, the Cyberlogic OPC Server Configuration Editor can automatically detect all devices attached to the Network Connections and create corresponding Network Nodes in the server configuration file.

A user can define many Network Connections, each having many Network Nodes. Later, the user will define the Access Paths used to read and write each of the Data Items. A user does not specify the network parameters for each Access Path. Instead, the Access Path simply references the Network Node, which already contains this information. Multiple Access Paths can refer to the same Network Node, greatly simplifying the configuration process. The network parameters are entered only once for each Network Node and any changes made to a Network Node are automatically reflected in all the Access Paths that reference that node.

## *Health Watchdog*

The Server can monitor the health of the connection to each physical device. If there is no network activity for a specified amount of time, the Server sends a communication request to the device to verify that it can still communicate. If the device becomes inaccessible, the Server rechecks it at a specified polling rate to see if it becomes accessible again. Once a failed Network Connection is reestablished, the Server continues to exercise the connection for a specified time to ensure that the connection is reliable. After these tests complete successfully, the node is marked as healthy again.

# Address Space Tree

The Address Space tree allows you to organize the Data Items in a way that makes sense for your project. You can group and name related Data Items regardless of where they exist in the physical devices.

The branches of the tree are Device Folders, Devices and Folders. These establish how the Data Items are organized. The Data Items themselves are the “leaves” of the tree. You will begin construction of the tree at the Address Space root folder, which may contain Device Folders and Devices.

## *Device Folders*

A Device Folder logically groups Devices and other Device Folders. You can place a Device Folder directly in the Address Space root folder or in another Device Folder, up to four levels deep.

## *Devices*

A Device in the Address Space tree represents a logical data source, which is associated with one or more Network Nodes to which the Server communicates. Each Device maintains a list of Access Paths and a list of Unsolicited Message Filters, which establish its relationship with the configured Network Nodes and Network Connections.

| | |
|---|---|
| **Note:** | A Device in the Address Space is not the same thing as a Network Node. Each Network Node represents a single physical device, while an Address Space Device may be associated with many physical devices. |

The main function of a Device is to define valid sources of data for all of its Data Items. Multiple Devices can use the same Network Node as a valid data source allowing greater flexibility in the logical grouping of Data Items.

You may place a Device directly in the Address Space root folder or in a Device Folder. In addition to its device-specific functionality, a Device operates as a normal Folder. It can contain Folders and Data Items.

## Access Paths

An Access Path is a logical connection to a Network Node. These connections link the Data Items in an Address Space Device with their values in a physical device. They tell the Server where and how to obtain these values during solicited data reads and writes.

| | |
|---|---|
| **Note:** | Access paths are required for solicited communications. Only the driver agents that support solicited communications support Access Paths |

Each Device in the Server's Address Space has a list of associated Access Paths. The Access Path at the top of the list is the primary Access Path; the rest are backups. If the current Access Path fails, the Server switches to the highest available backup. This feature allows you to set up redundant networks for greater communication reliability. In addition, if your controls design uses a backup controller, you can set up Access Paths to both the primary controller and its backup.

When an Access Path fails, the Server monitors it to see if it becomes accessible again. Once a failed Access Path becomes available, the Server continues to exercise the connection for a specified delay time to ensure that the connection is reliable. After these tests complete successfully, the Server switches back to the higher priority Access Path.

## Unsolicited Message Filters

In addition to the more common solicited updates, the Cyberlogic OPC Server also supports unsolicited data updates. In a solicited update, the Server sends a request to a device asking it for data, and the device replies. In an unsolicited update, the device decides when to send data to the Server. This helps to minimize the amount of traffic on the network. For example, instead of having the Server poll a device every 500 milliseconds to see if some data has changed, you can configure the device to update the Server only when the data changes.

The disadvantage of unsolicited updates is the fact that the Server has no control over who may decide to send data to it. Data written from unintended sources could corrupt the Server-maintained data, resulting in potentially catastrophic events.

Although unrestricted unsolicited updates are possible, the Cyberlogic OPC Server supports a mechanism of unsolicited message filters to prevent data corruption. Unsolicited messages must first pass through the user-defined filters before the Server accepts them. User-defined filters guarantee the unsolicited messages are accepted only from trusted sources.

| | |
|---|---|
| **Note:** | Unsolicited message filters are used only for unsolicited communications. Only the driver agents that support unsolicited communications support unsolicited message filters. |

The unsolicited message filters are organized in groups. Each group has an equal priority and a message must pass through at least one of these groups to be accepted by the Server.

### Unsolicited Message Filter Groups

An unsolicited message filter group is a list of trusted Network Nodes and trusted Network Connections. It supports two modes of operation. In the default mode, the Server accepts messages that pass any of the configured filters. In the alternative Priority Unsolicited mode, the Server treats the filter list as a ranked list of preferred and backup data sources. It monitors the connections to each unsolicited message source and accepts messages only from the highest ranked node that has a healthy connection.

## *Folders*

Folders logically group Data Items and other Folders. A Folder can be placed directly under a Device or under another Folder, up to four levels deep.

## *Data Items*

A Data Item represents a register in the physical device, a range of registers, a bit inside a register or a range of bits. The user can individually configure each Data Item for solicited updates, unsolicited updates or both.

The Cyberlogic OPC Server supports a number of integer, floating point and string data types. It also supports single-dimensional arrays of these types. The following table shows all supported simple data types:

| Data Type | Data Size (bits) | Default Canonical Data Type | Description |
|---|---|---|---|
| Default | | | Default type based on the data item address |
| BIT | 1 | VT_BOOL | 1-bit boolean |
| SINT8 | 8 | VT_I1 | Signed 8-bit integer |
| UINT8 | 8 | VT_UI1 | Unsigned 8-bit integer |
| SINT16 | 16 | VT_I2 | Signed 16-bit integer |
| UINT16 | 16 | VT_UI2 | Unsigned 16-bit integer |
| SINT32 | 32 | VT_I4 | Signed 32-bit integer |
| UINT32 | 32 | VT_UI4 | Unsigned 32-bit integer |
| SINT64 | 64 | VT_I8 | Signed 64-bit integer |
| UINT64 | 64 | VT_UI8 | Unsigned 64-bit integer |
| FLOAT32 | 32 | VT_R4 | IEEE format 32-bit floating point number |
| FLOAT64 | 64 | VT_R8 | IEEE format 64-bit floating point number |
| BCD16 | 16 | VT_UI2 | BCD value in the range of 0 - 9999 |
| BCD32 | 32 | VT_UI4 | BCD value in the range of 0 - 99999999 |
| STRING | String size * 8 | VT_BSTR | Zero terminated ASCII string of 8-bit characters |
| WSTRING | String size * 16 | VT_BSTR | Zero terminated UNICODE string of 16-bit characters |
| FIELD | Field size | The best fitting VT_UIx or array of VT_UI1 if size > 64 | Multiple bit field |

For each simple data type, a user can select a specific canonical data type (a variant data type in the form of VT_XXX) or choose the default type. When the default type is selected, the Server selects the appropriate canonical data type that can best store the selected data type inside the Server.

### Data Write Protection

In general, the Cyberlogic OPC Server supports read and write operations to its Data Items. However, writing to some Data Items may create a safety hazard. Some registers, such as PLC-5 Inputs, are read-only and require no additional protection. For read/write registers, you can disable the write capability at any level. That is, you can disable writes for a:

- Data Item
- Folder
- Device
- Device Folder
- Network Node
- Network Connection
- Driver Agent

You can also disable DirectAccess writes at each Network Node, Network Connection or driver agent.

## *DirectAccess*

At run time, in addition to the user-configured Address Space branches, the Cyberlogic OPC Server dynamically creates a branch called DirectAccess at the root of the Address Space. OPC clients can use this branch to access any register in any configured Network Node by directly specifying the register address.

The structure of this branch depends upon the driver agent and the configuration of the Network Connections. The DirectAccess branch acts like a Device Folder that contains all configured driver agents. Each driver agent branch contains its configured Network Connections, and each Network Connection branch contains its configured Network Nodes. However, only driver agents, Network Connections and Network Nodes that enable DirectAccess are present.

DirectAccess can benefit users in two ways. First, you can deploy minimally configured servers quickly, giving clients access to data in the shortest possible time. By configuring just the Network Connection and Network Nodes, a user would have access to all the registers in each Network Node.

Second, DirectAccess can help you to work around configuration errors. Suppose a user forgets to configure a needed data register in the Server. DirectAccess allows an OPC client to access the forgotten register until the Server configuration is modified.

# Conversions

The raw data associated with a Data Item may represent a signal value from some instrument. In most cases, this value is not expressed in the engineering units of the measured signal. To simplify operating on the signal's data, the Cyberlogic OPC Server can associate a Conversion with each Data Item.

A user can define a number of different types of Conversions. The Server can then apply each Conversion to a number of Data Items. As a result, a user need not define the same Data Conversion for each Data Item.

The Server supports both linear and square root Conversions. Each has a range of engineering units that corresponds to the specified instrument range. The Linear and Square Root conversions keep a linear or square root relation between the engineering units range and the instrument range.

In addition, the Server supports data range clamping. The clamping can be based upon either the engineering units range or a custom range.

# Simulation Signals

To facilitate client side testing without the need for a physical device, a predefined formula can simulate the data for each Data Item. A user can define several different types of Simulation Signals. Each signal can then simulate a number of Data Items. As a result, a user need not define the same Simulation Signal for each Data Item.

The Simulation Signals available are: read count, write count, random, ramp, sine, square, triangle and step. The signals other than read count and write count have parameters that define properties such as amplitude, signal phase and number of steps.

Data can be simulated at any level in the Server Address Space. Enabling data simulation at one level automatically enables it at all levels below. This allows you to switch quickly between simulated and real data for a large number of Data Items.

# Alarms and Events

The Cyberlogic OPC Server supports the OPC Alarms and Events specification. It allows a user to define a number of alarm conditions, each of which can then be used by a number of Data Items. As a result, a user need not define the same alarm condition for each Data Item. Alarms cannot be used with string data items, arrays or bit fields greater than 64 bits. There are two categories of alarms: digital and limit (analog).

| | |
|---|---|
| Note: | To receive the alarms and events reported by the server, the client application must also support the OPC Alarms and Events specification. |

## Limit Alarms

Limit alarms are normally used with numeric data. These alarm definitions divide the Data Item range into five alarm states: LoLo, Lo, Normal, Hi and HiHi.

Every alarm state includes an alarm message and a severity level. In addition, you may indicate whether the alarm requires a client-side acknowledgement. An optional deadband value prevents the Server from generating a large number of alarm messages when the signal oscillates around one of the limits. When the deadband value is set properly, the Server will send only one alarm even if the signal oscillates.

## Digital Alarms

Digital alarms are normally used with Boolean Data Items. A user can request an alarm when the item's value equals either TRUE or FALSE.

Each alarm has an associated alarm message and a severity level. The alarm message describes the alarm condition. The severity value ranges from 1 to 1000 and indicates the importance of the alarm. Optionally, an alarm can be generated when the item's data returns to its normal value. A user can also specify that each alarm condition requires a client-side acknowledgement.

# Server Status Block

The Cyberlogic OPC Server has a set of 16-bit registers called the *Server Status Block*. Physical devices on the network, such as PLCs, can read these registers to determine the current health and status of the Server. The location of the Server Status Block depends upon the network type. For example, in the DHX OPC Server, the Server Status Block is laid-out as follows:

| PLC-2 | Other PLCs | Description |
|---|---|---|
| 200 | N7:0 | Server version – Major (e.g. 5) |
| 201 | N7:1 | Server version – Minor (e.g. 0) |
| 202 | N7:2 | Server version – Build (e.g. 3) |
| 203 | N7:3 | User configured server signature |
| 204 | N7:4 | Server alive millisecond counter (Low word) |
| 205 | N7:5 | Server alive millisecond counter (High word) |

For information about location of the Server Status Block for a particular driver agent, refer to the help file specific for that Agent.

# On-Line Configuration Changes

You can make and apply configuration changes on-line, while OPC clients are connected to the server. A user has full control over when these changes are applied: once following several changes or after each configuration change. When the changes are applied, all connected clients are updated with the new configuration.

# Undoing Configuration Changes

The Cyberlogic OPC Server Configuration Editor keeps track of recent configuration changes. Before saving the changes, a user can revert to the previously saved configuration.

# Configuration Import/Export

To speed up the configuration of multiple similar servers, the Cyberlogic OPC Server Configuration Editor has an Import/Export capability. An entire database can be exported and then re-imported or just selected portions of it can be imported.

The utility can export to and import from many different file formats, including imports of the configuration files of other OPC server vendors.

# Data Monitor

The Cyberlogic OPC Server Configuration Editor includes a simple tag monitoring utility. This allows a user to quickly test changes to the Server's configuration. It can also aid in troubleshooting.

# CONFIGURATION

You must properly configure the Cyberlogic OPC Server before you can use it. To do this, you must run the Cyberlogic OPC Server Configuration Editor after the installation.

This section is divided into three subsections:

- Typical Configuration Session is a step-by-step tutorial that is a good place to start for a first-time user who needs an overview of the configuration process.
- Creating Network Connections and Nodes shows how to add a network or control device to your system. This would be a good place to go if you are adding hardware to an existing configuration.
- OPC Server Configuration Editor provides detailed information on the capabilities of the configuration editor, including simulation, conversions and alarms.

For detailed information on configuring access to specific families of control devices, refer to the help file for the driver agent you are using.

## Typical Configuration Session

The following steps show a typical configuration session using the DHX driver agent. Configurations using the other driver agents, such as MBX and ControlLogix, would be very similar. Only the most common features are shown here, so you should use this description only as a guideline. For detailed information on the features of the Configuration Editor, refer to the help file for the driver agent you are using.

The first step in configuring the Cyberlogic OPC Server is to create at least one DHX device. Depending upon your communication network, you must have at least one of the following Cyberlogic driver products installed:

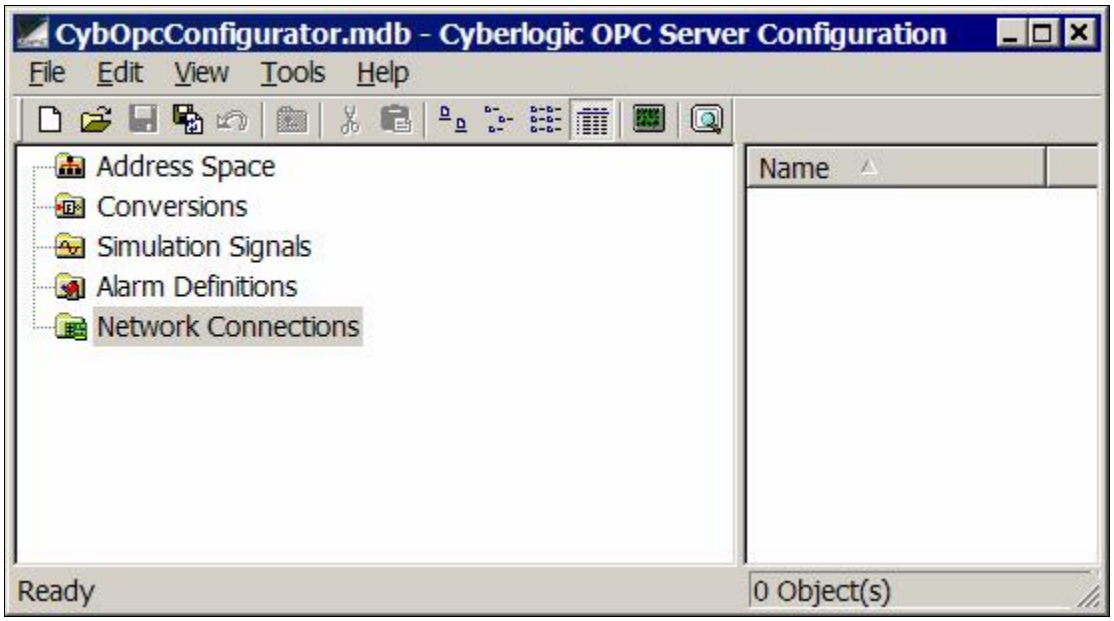
- DHX Driver
- Ethernet DHX Driver
- Serial DHX Driver
- DHX Gateway Driver

This tutorial assumes that you are running under Windows XP and have a single 1784-KTX card connected to the Data Highway Plus (DH+) network. If you had used a Plug-and-Play PCI card, such as the 1784-PKTX/A a DHX device would have been created when you booted up the system. Because the 1784-KTX is not Plug-and-Play, you must create the device manually. If you use a different adapter card or different network, refer to the driver-specific help file for more information on configuring the DHX devices.

Another assumption is that you have two Allen-Bradley PLC-5/20s connected to the Data Highway Plus network. One is the primary PLC at node address 2 while the other one is the backup PLC at node address 50.

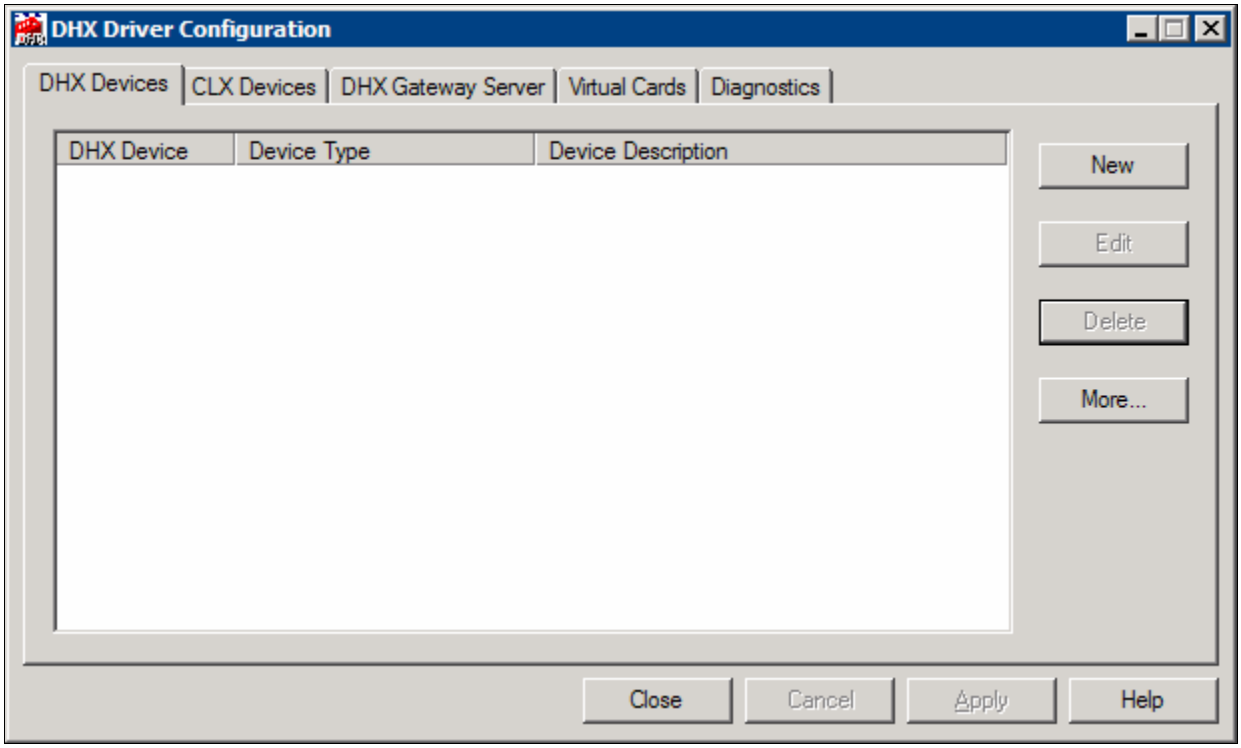
1. To start the editor, open the Windows Start menu, locate the DHX OPC Server submenu and select the *OPC Server Configuration* menu item.

   You will see the following screen:

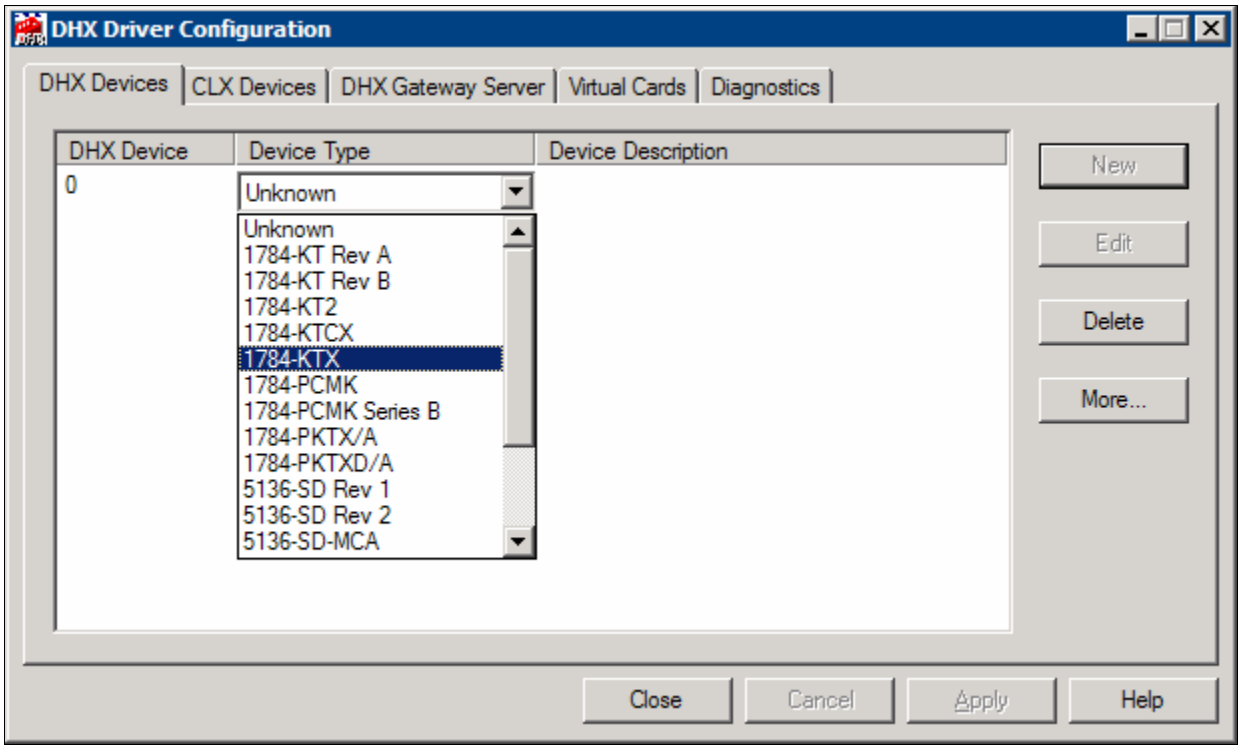



   Since you are running the Cyberlogic OPC Server Configuration Editor for the first time, the editor will prompt you to go to the File menu and open a new configuration database. You will start with an empty database. The first step is to configure the KTX device.

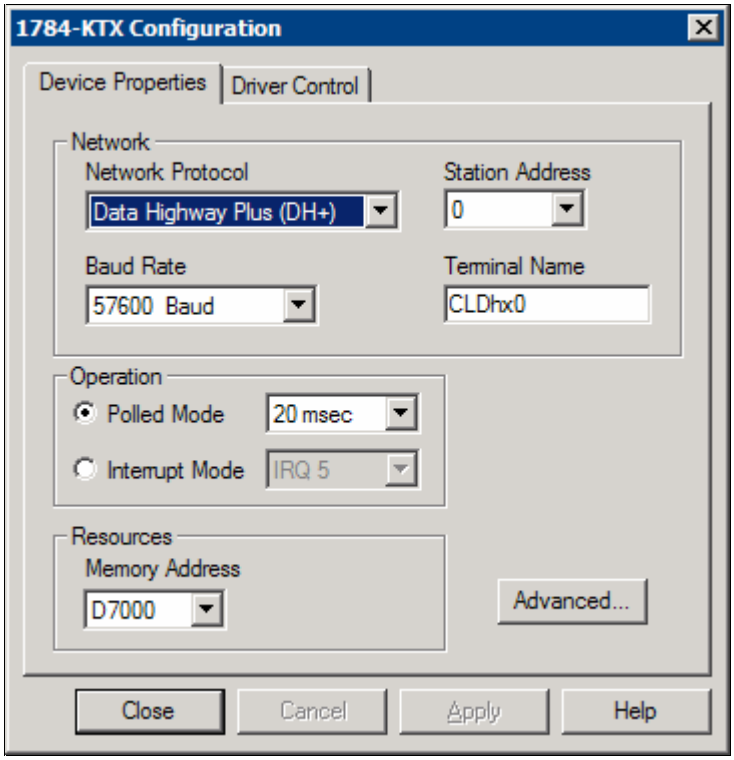
2. From the Tools menu select *DHX Connections for Allen-Bradley,* and then select *DHX Driver Configuration*…. You will see the following screen:



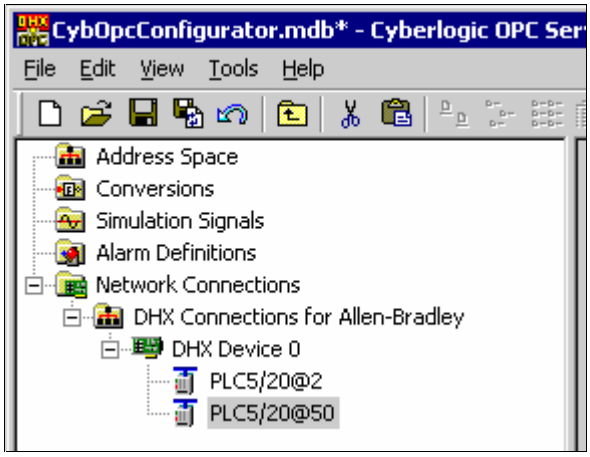
3. Click the *New* button and select *1784-KTX* from the drop-down box.




4. The configuration window shown below pops up. Select the correct memory address and then click *OK* to return to the DHX Driver Configuration dialog. Click *Close*.



5. Select the *Network Connections* root folder and select *Auto Config* from the Edit menu (or right-click on the *Network Connections* root folder and select *Auto Config* from the context menu). The editor will try to find all Network Connections and automatically detect and configure all Network Nodes. When it is done, you will see the following configuration in the Network Connections tree:



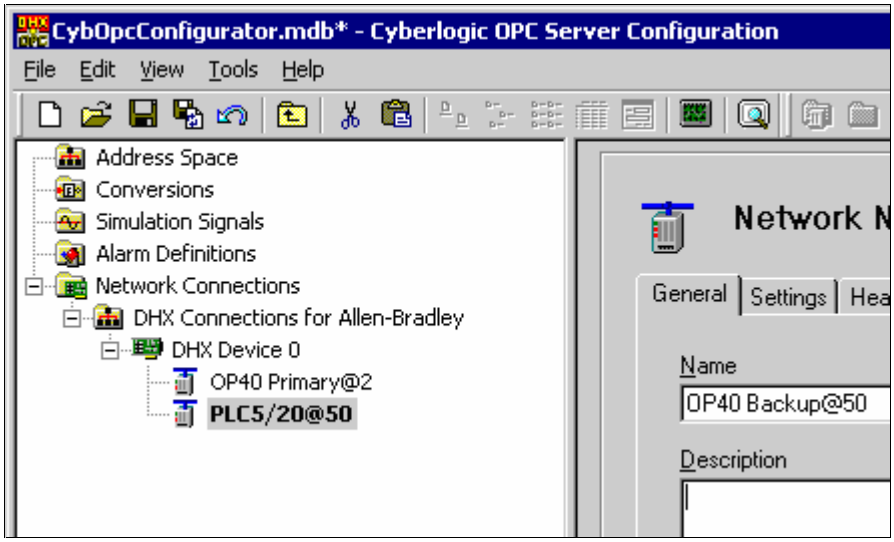

This screen shows that the editor has detected two programmable controllers, one at node address 2 (*PLC-5/20@2*) and the other one at the node address 50 (*PLC-5/20@50*).

| **Note:** | ControlLogix nodes do not report enough information to permit Auto Config to identify them. They will be detected, but will be reported simply as DHX nodes of unknown type. |
|---|---|

You have now completed the minimum required configuration for the server. You need not do any additional configuration if you limit yourself to DirectAccess of the PLC registers. The rest of this tutorial will highlight additional features of the server.

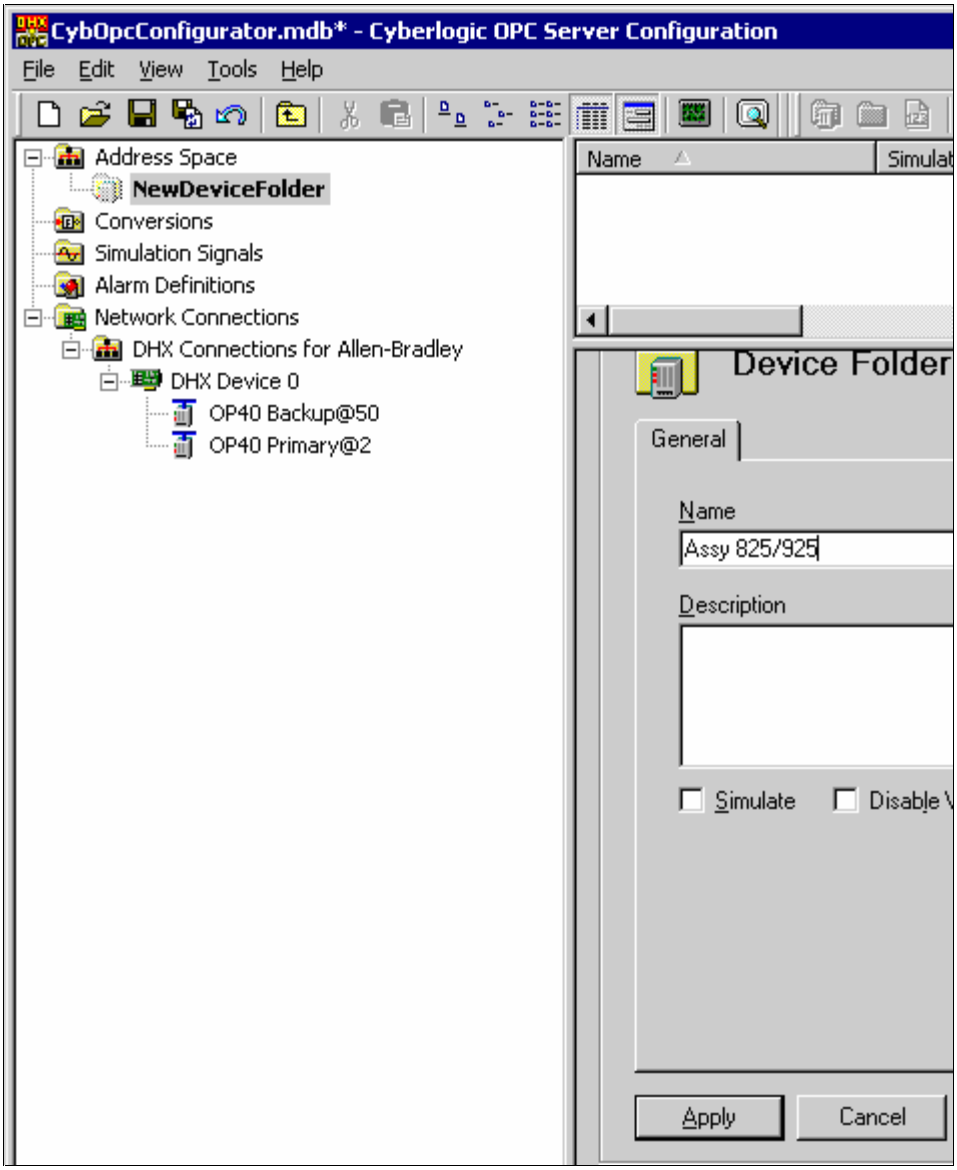
Before doing this, modify the default network node names to make them more descriptive.

6. Select the *PLC-5/20@2* network node. In the Name field of the General tab, change the name to *OP40 Primary@2*. Then select the *PLC-5/20@50* network node. Change the name to *OP40 Backup@50*.



Before moving on to configuring the server's Address Space tree, you may want to check the settings for the detected Network Connection and the Network Nodes. For most users there will be no need to make any changes to the auto-detected settings.

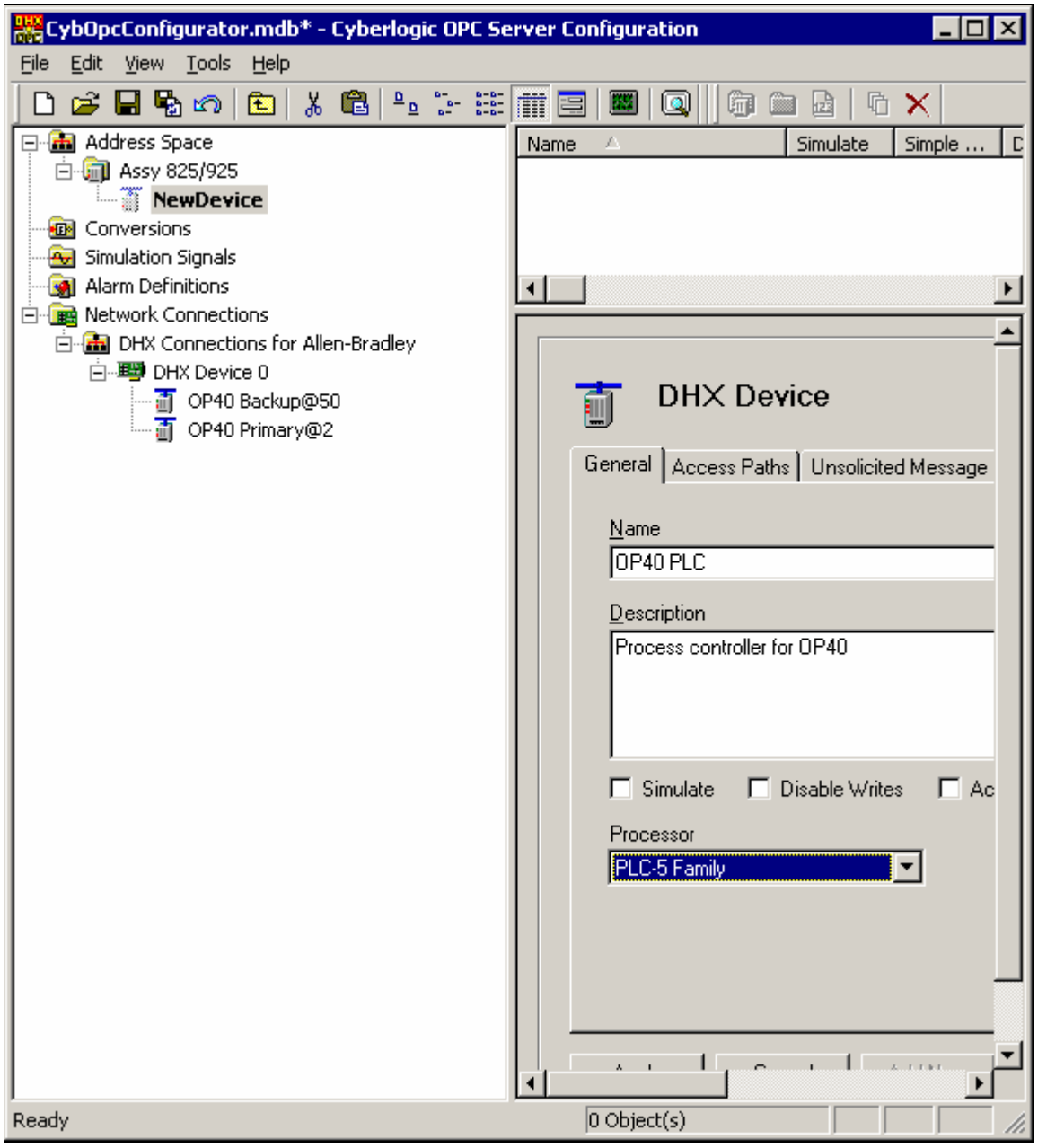
7. Select the *Address Space* root folder and then select *New/Device Folder* from the Edit menu (or right-click the *Address Space* root folder and select *New/Device Folder* from the context menu). The folder's name defaults to *NewDeviceFolder*. Enter a unique name (for this exercise, use the name *Assy 825/925*) and click the *Apply* button.




You have just created a Device Folder. A Device Folder logically groups Devices and other Device Folders. You can place a Device Folder directly under the Address Space root folder or inside another Device Folder, up to four levels deep.

8. Select the newly created *Assy 825/925* device folder. From the Edit menu select *New/Device/DHX Connections for Allen-Bradley* (or right-click on *Assy 825/925* and select *New/Device/DHX Connections for Allen-Bradley* from the context menu). The device name defaults to NewDevice. Enter a unique name (for this exercise, use the name *OP40 PLC*) and an optional description

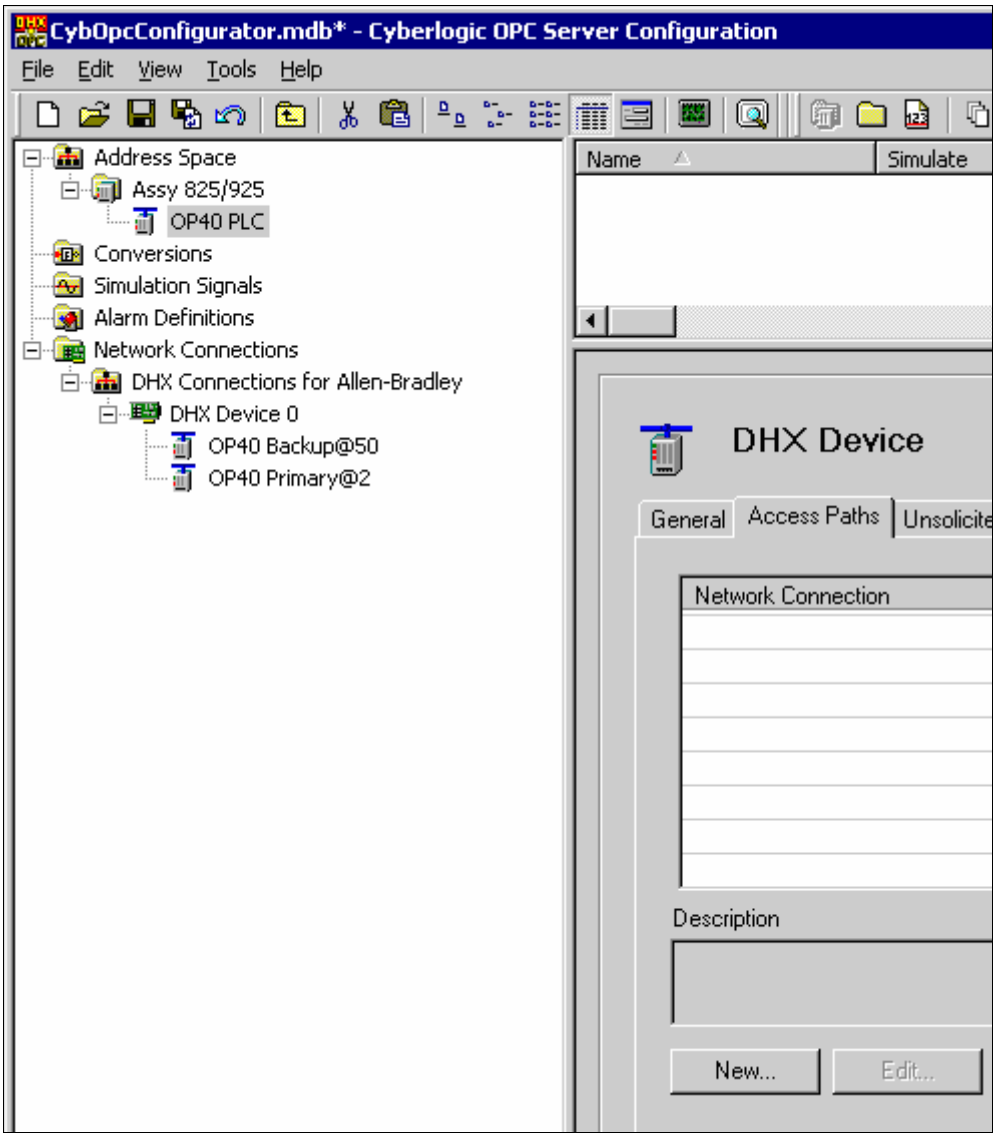
(*Process controller for OP40*). From the *Processor* drop-down box, select the controller type (*PLC-5 Family*) and click *Apply* when done.



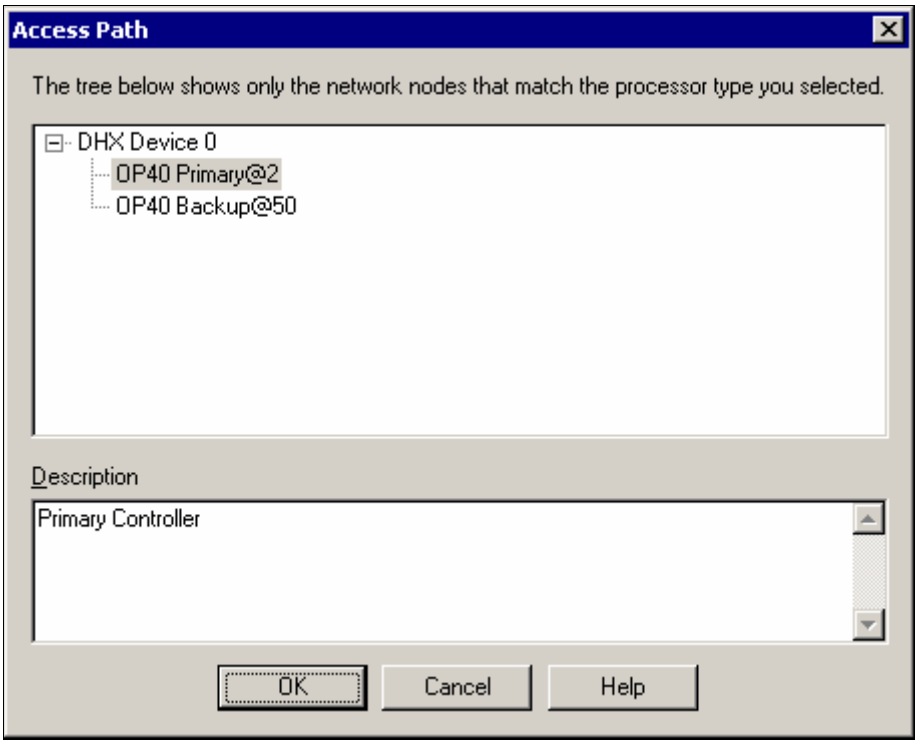

You have just created a Device. A Device in the Address Space tree represents a logical data source, which is associated with one or more physical devices to which the Server communicates. You can place a device directly in the Address Space root folder or in a Device Folder. In addition to its device-specific functionality, a Device operates as a Folder. It can contain Folders and Data Items.

The newly created OP40 PLC Device has not been associated with the physical Network Nodes yet. In the following steps, you will set OP40 Primary@2 as the primary controller and OP40 Backup@50 as the backup controller.

9. Select the *Access Paths* tab and click the *New…* button.

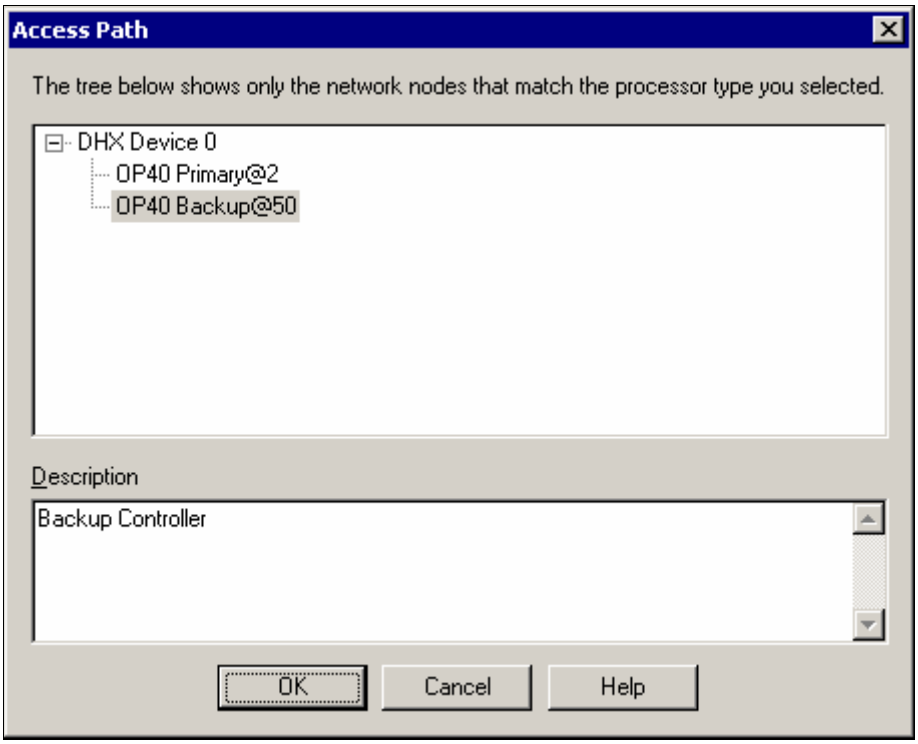


10. In the Access Path dialog, select the *OP40 Primary@2* node under DHX Device 0. Enter *Primary Controller* in the optional description field and then click *OK*.

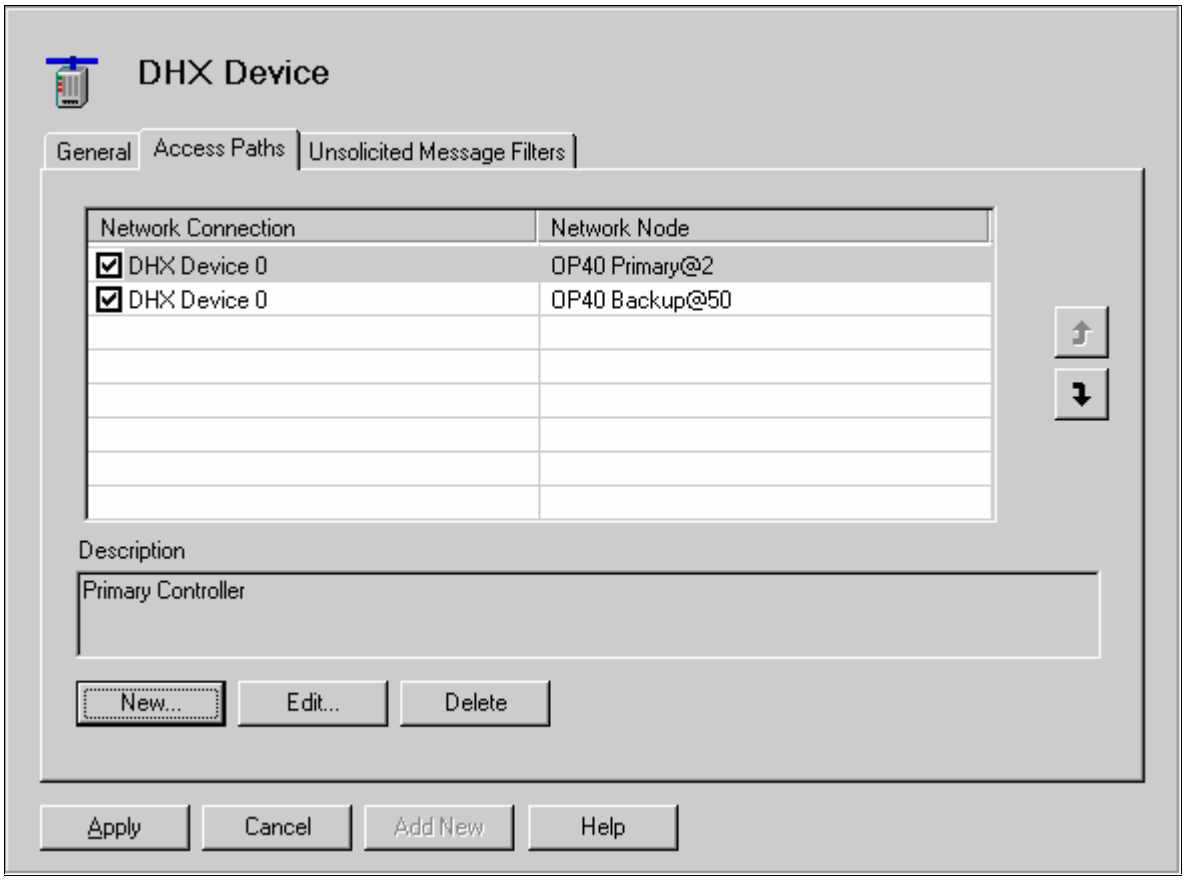



You have just created the primary Access Path. Now you will add a backup Access Path that will be used in case the primary connection fails.

11. Click the *New…* button. In the Access Path dialog, select the *OP40 Backup@50* node under DHX Device 0. Enter *Backup Controller* in the optional description field.



12. When you have finished entering the information, click *OK*. The Access Path configuration will look like the screen below.



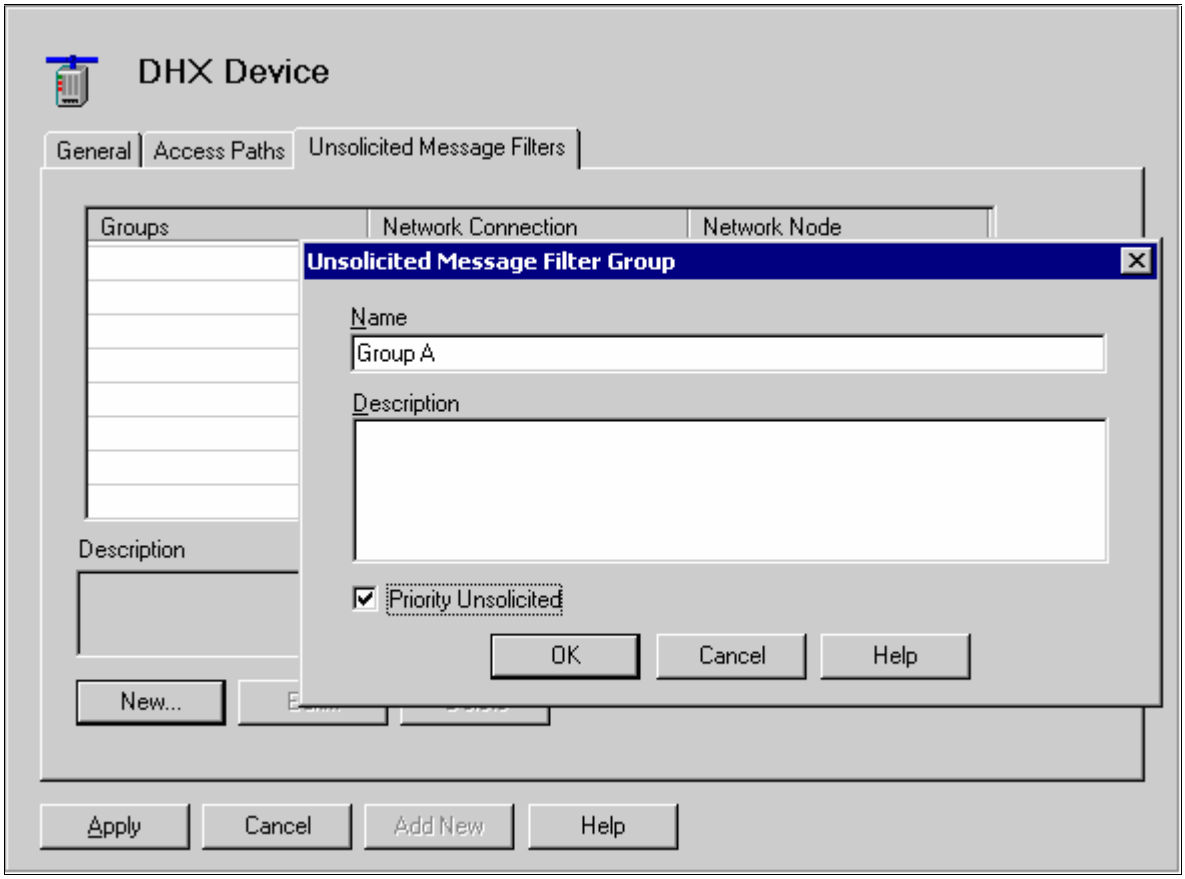

You have now created two alternative Access Paths, one to the primary PLC and one to the backup PLC. The order of the paths shown on the screen is also the priority order. The Server will use the first path for solicited communications, if it is working. If that Access Path fails, the Server will switch to the next Access Path on the list. The Server will continuously monitor the failed Access Path so that it can switch back to it when the primary Access Path becomes available. To change the priority order for the Access Paths, select an Access Path and use the arrow keys to the right of the window to move that path up or down the list.

You may specify an unlimited number of alternative Access Paths. For example, if you have two alternative network connections for each PLC, perhaps Data Highway Plus and Ethernet, you can create four Access Paths: Data Highway Plus and Ethernet Access Paths for the primary PLC, and Data Highway Plus and Ethernet Access Paths for the backup PLC. Again, the order in which these Access Paths are placed would specify the order in which the Server would switch to the backup connections.

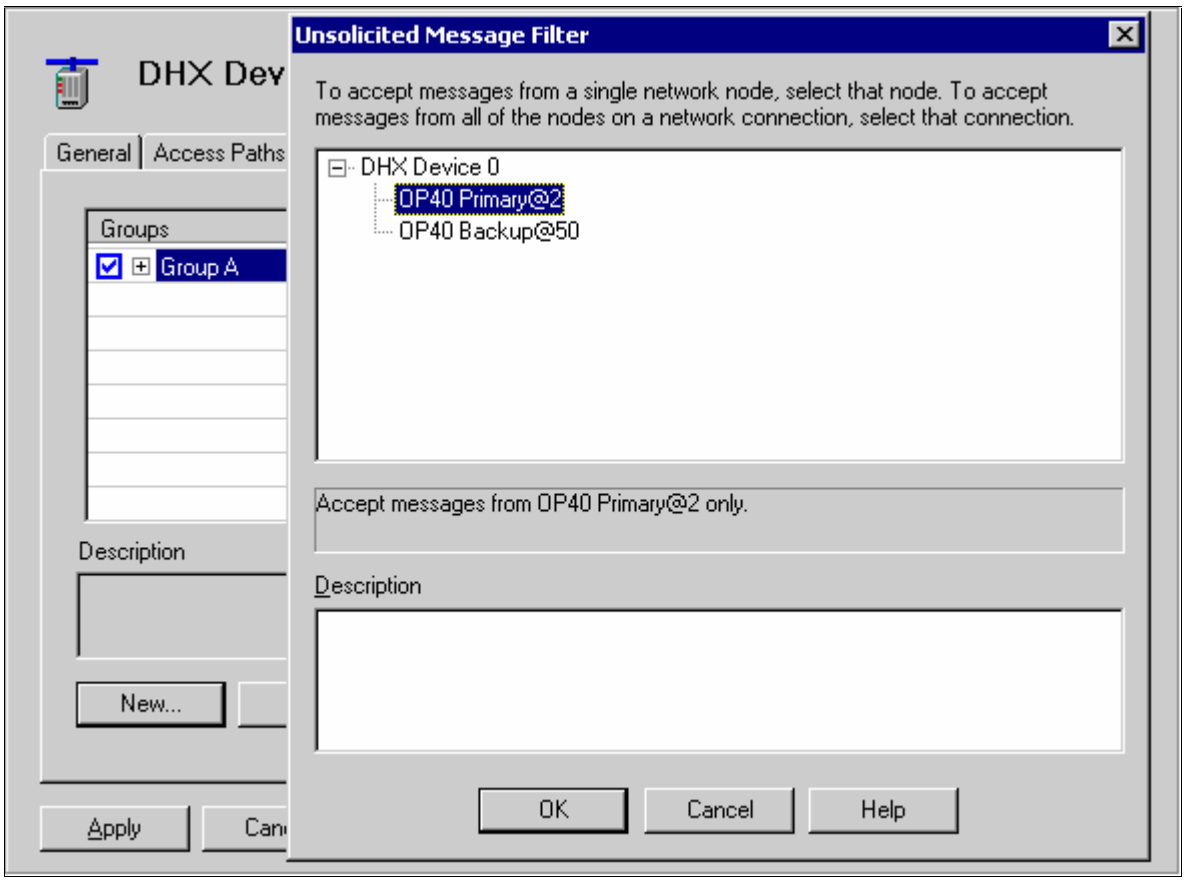
Next, you will configure the unsolicited message filters. Unsolicited messages must pass through these user-defined filters before they are accepted. The filters help to ensure that the Server accepts unsolicited messages only from trusted sources.

13. Select the *Unsolicited Message Filters* tab. Click the *New…* button and select *Group…* from the context menu. You will organize the filters you create into groups, which may be prioritized or non-prioritized. In addition, the Configuration Editor allows you to disable and enable entire groups of filters. This can be very convenient during startup and debugging.

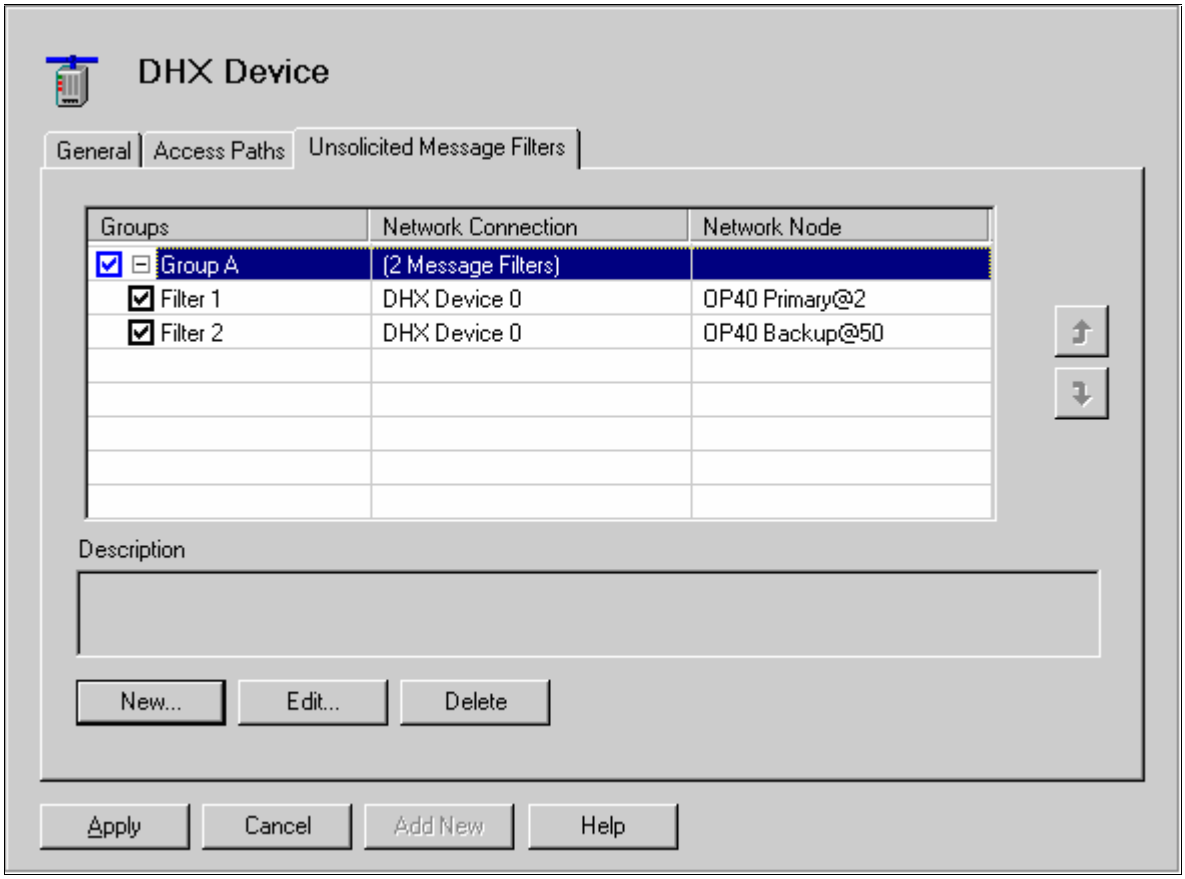
14. In the Unsolicited Message Filter Group dialog, enter the name of the Filter Group (for this exercise, use *Group A*) and an optional description, if you wish. Check the *Priority Unsolicited* box and click the *OK* button.



15. Click the *New…* button and select *Filter…* from the context menu. In the Unsolicited Message Filter dialog, select *OP40 Primary@2* and then click *OK*. This creates a filter that will accept unsolicited messages from the primary PLC.



16. Repeat the preceding step, this time selecting *OP40 Backup@50* to create a filter to accept unsolicited messages from the backup PLC. If you then click the + sign beside Group A, your filter configuration will look like this.




You have now created two alternative unsolicited message filters, one to the primary and one to the backup PLC. Because you marked the group *Priority Unsolicited*, the Server treats the filters within the group as a ranked list of data sources. It monitors the connections to each unsolicited message source and only accepts messages from the highest ranked node that has a healthy connection.

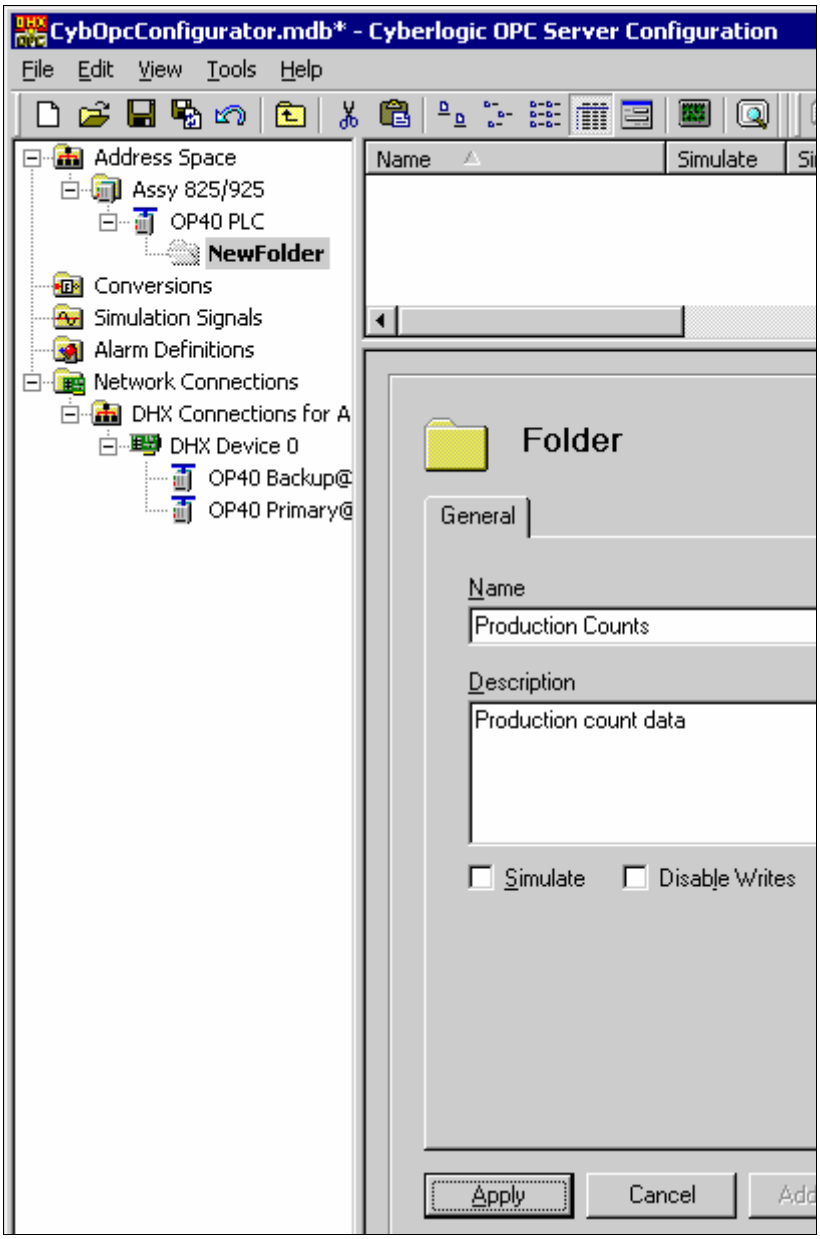
In this example, if the connection to the primary PLC is healthy, only messages from this PLC will be accepted. However, if the primary connection fails, only messages from the backup PLC will be accepted. Other network nodes are not allowed to send messages to this device.

If you had cleared the Group A Priority Unsolicited check box, the Server would treat all of the filters in the group equally. Any unsolicited message that passed any of the filters in the group would be accepted.

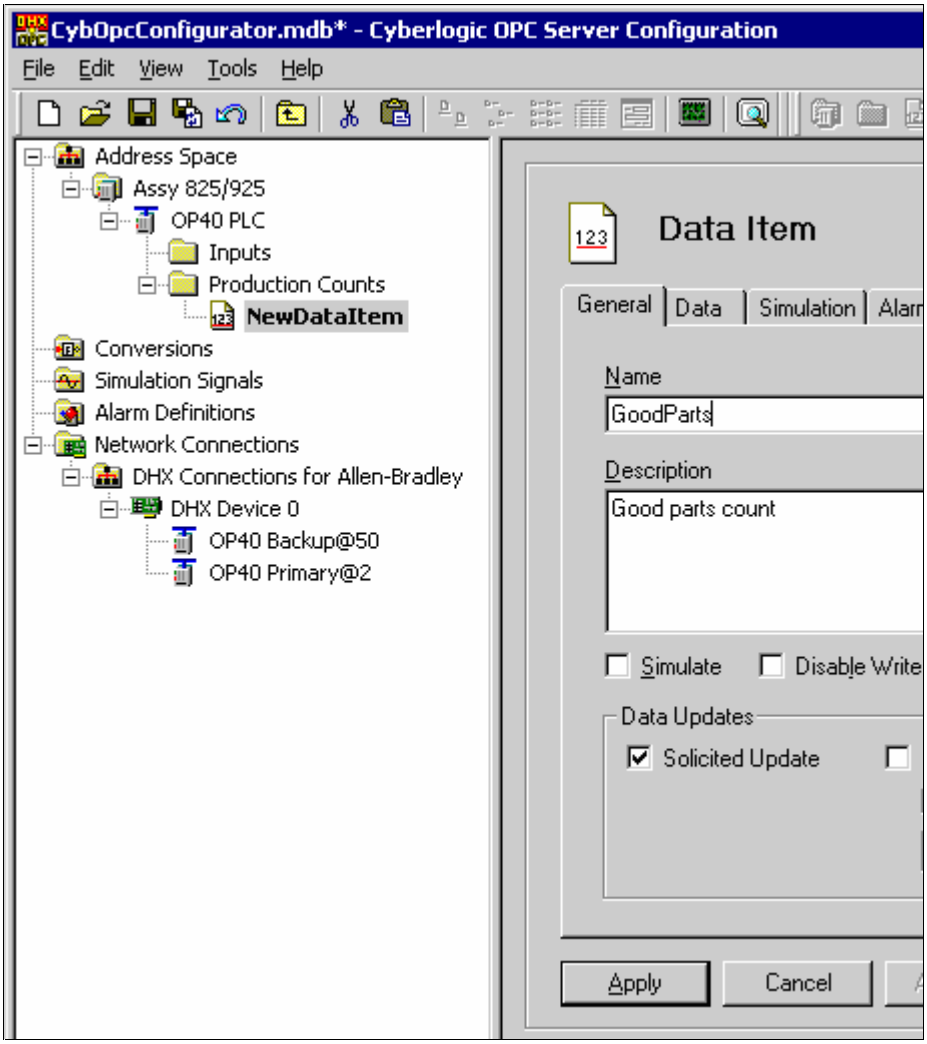
You can configure additional groups of filters. Each group can be marked as Priority Unsolicited or not, as your application may require. Notice that the priority property applies only to the filters within the group. There is no priority implied between the groups themselves.

Now that you have configured a device, the next step is to organize and configure the data items.

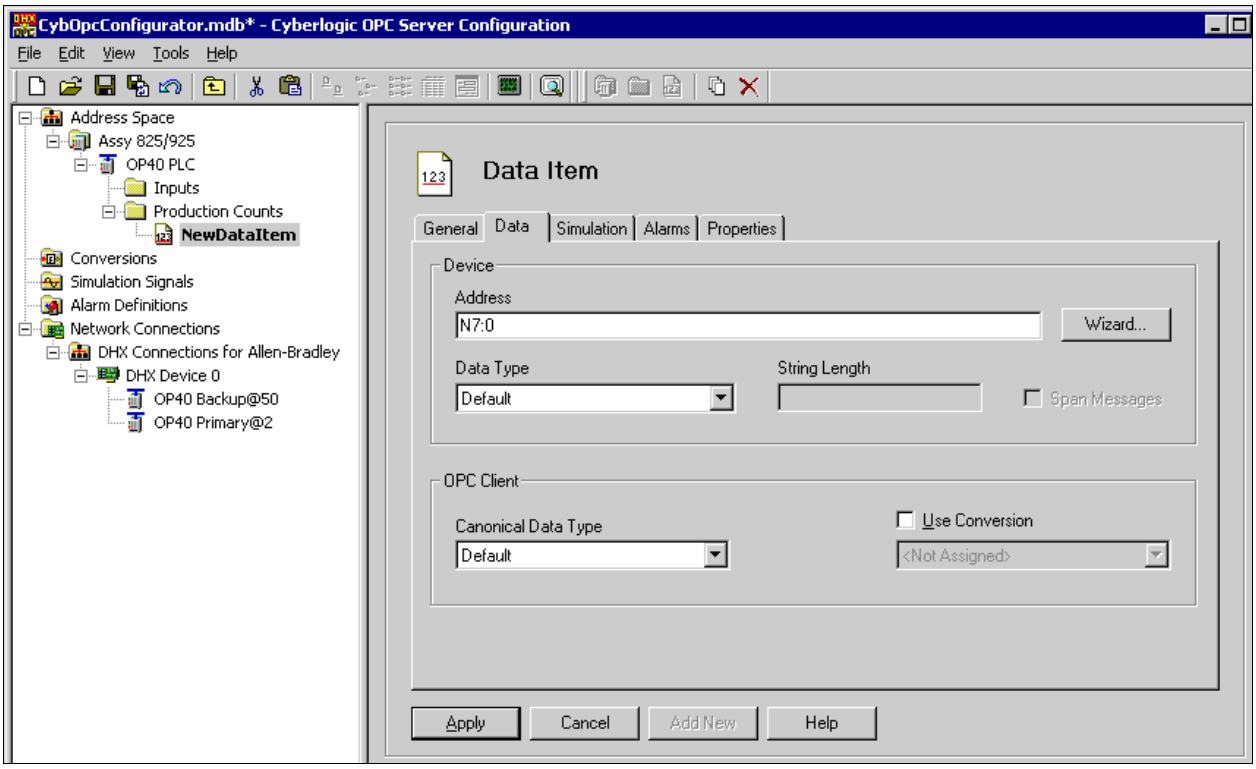
17. Select the *OP40 PLC* device. Now from the Edit menu select *New/Folder* (or right-click on *OP40 PLC* and select *New/Folder* from the context menu). Enter a unique name (in this example, *Production Counts*) and an optional description. When you have finished, click *Apply*. In the same way, create another folder called Inputs with the description Discrete inputs.


CybOpcConfigurator.mdb* - Cyberlogic OPC Server Configuration
File Edit View Tools Help
Address Space
Assy 825/925
OP40 PLC
NewFolder
Conversions
Simulation Signals
Alarm Definitions
Network Connections
DHX Connections for A
DHX Device 0
OP40 Backup@
OP40 Primary@
Name
Simulate
Folder
General
Name
Production Counts
Description
Production count data
Simulate
Disable Writes
Apply
Cancel

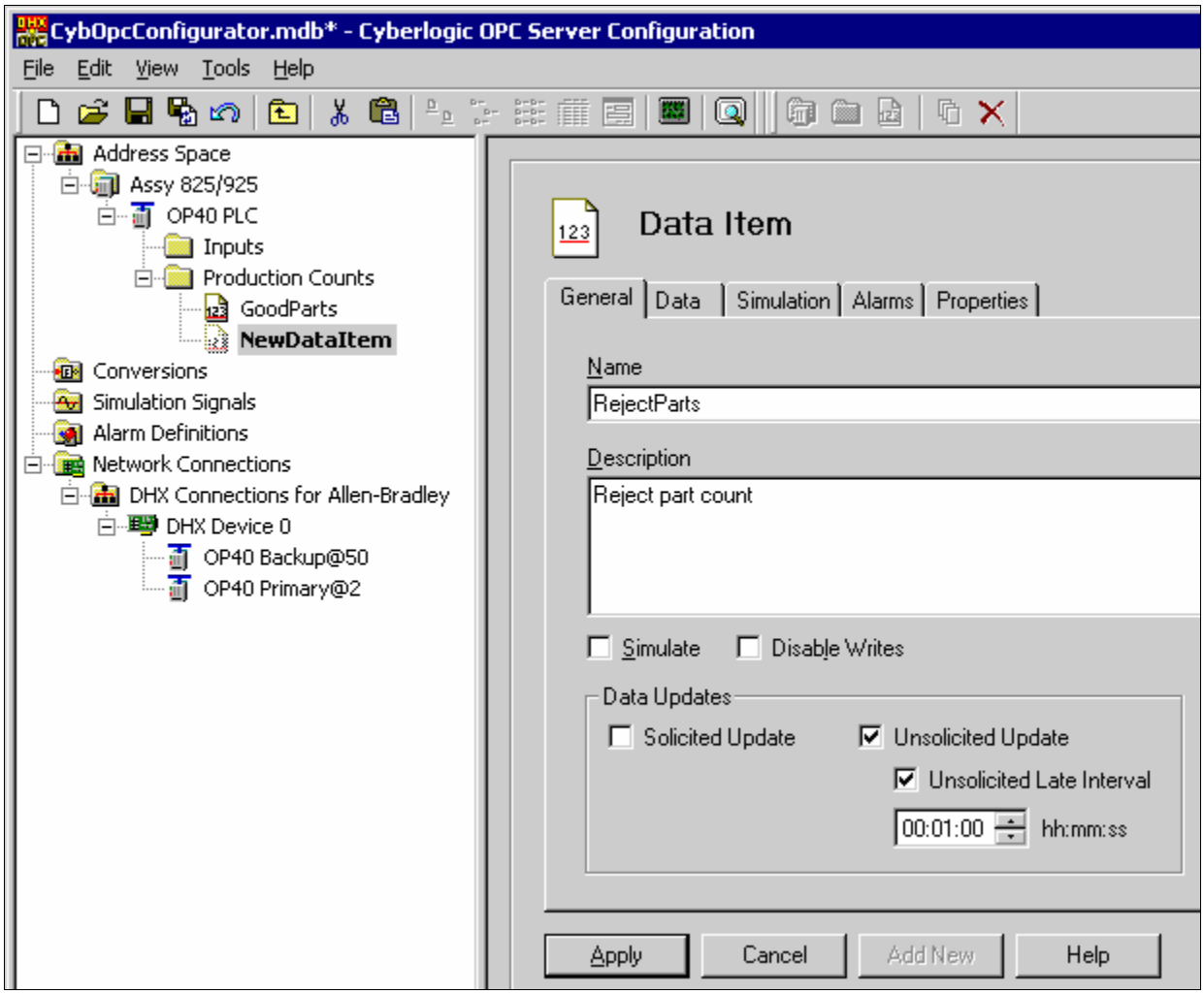
18. Select the *Production Counts* folder and select *New/Data Item* from the Edit menu (or right-click on *Production Counts* and select *New/Data Item* from the context menu). Enter a unique name (for this example, *GoodParts*) and an optional description (*Good parts count*). Notice that the Solicited Update box is checked.



19. Select the *Data* tab and keep the default address of *N7:0*. Click *Apply* to complete the Data Item creation.



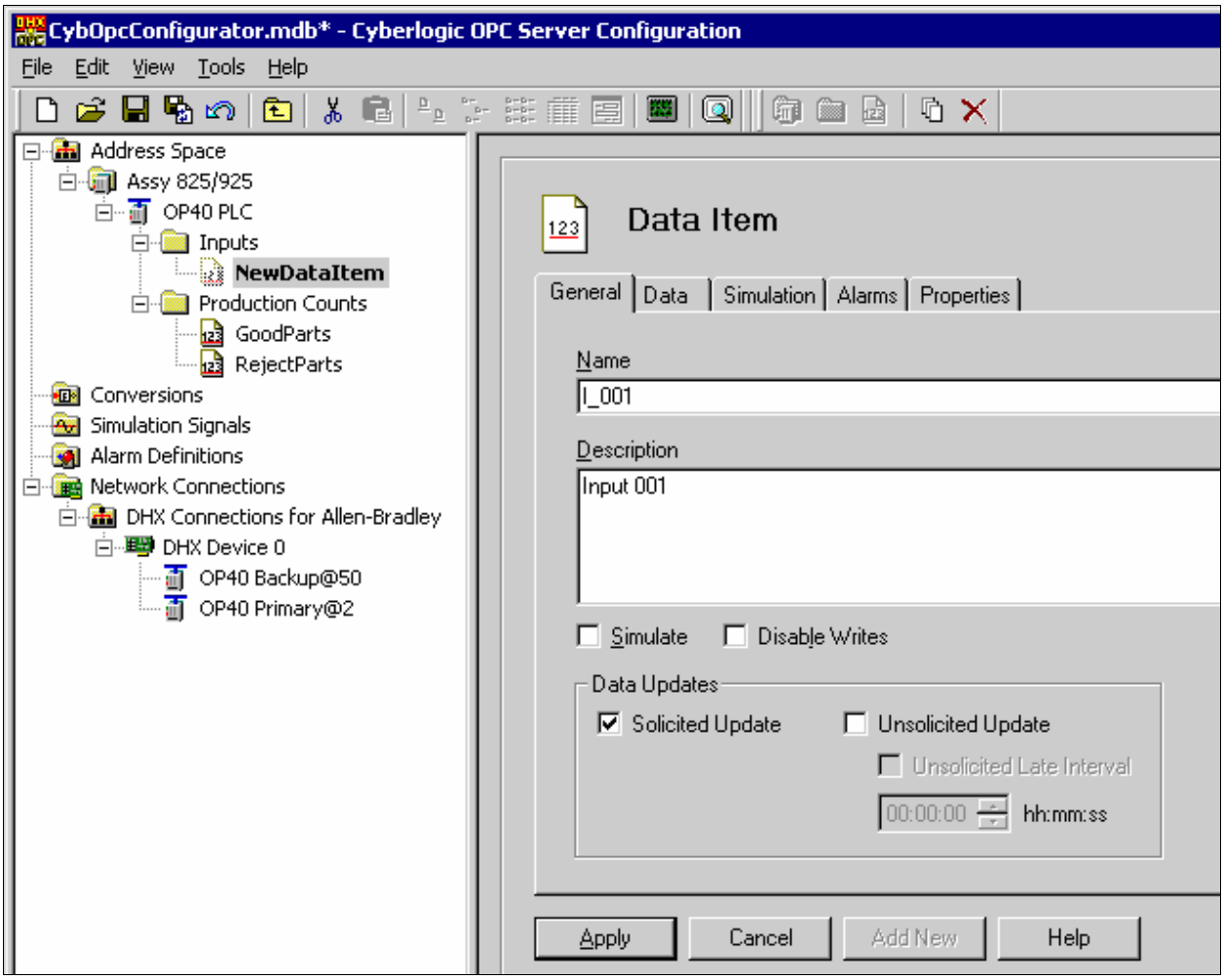
20. Repeat the process to create another Data Item called *RejectParts* at the register address N7:1. However, for this item, uncheck the default *Solicited Update*, check *Unsolicited Update* and check *Unsolicited Late Interval*. Set the Late Interval to 1 minute. Click *Apply* when you are done.



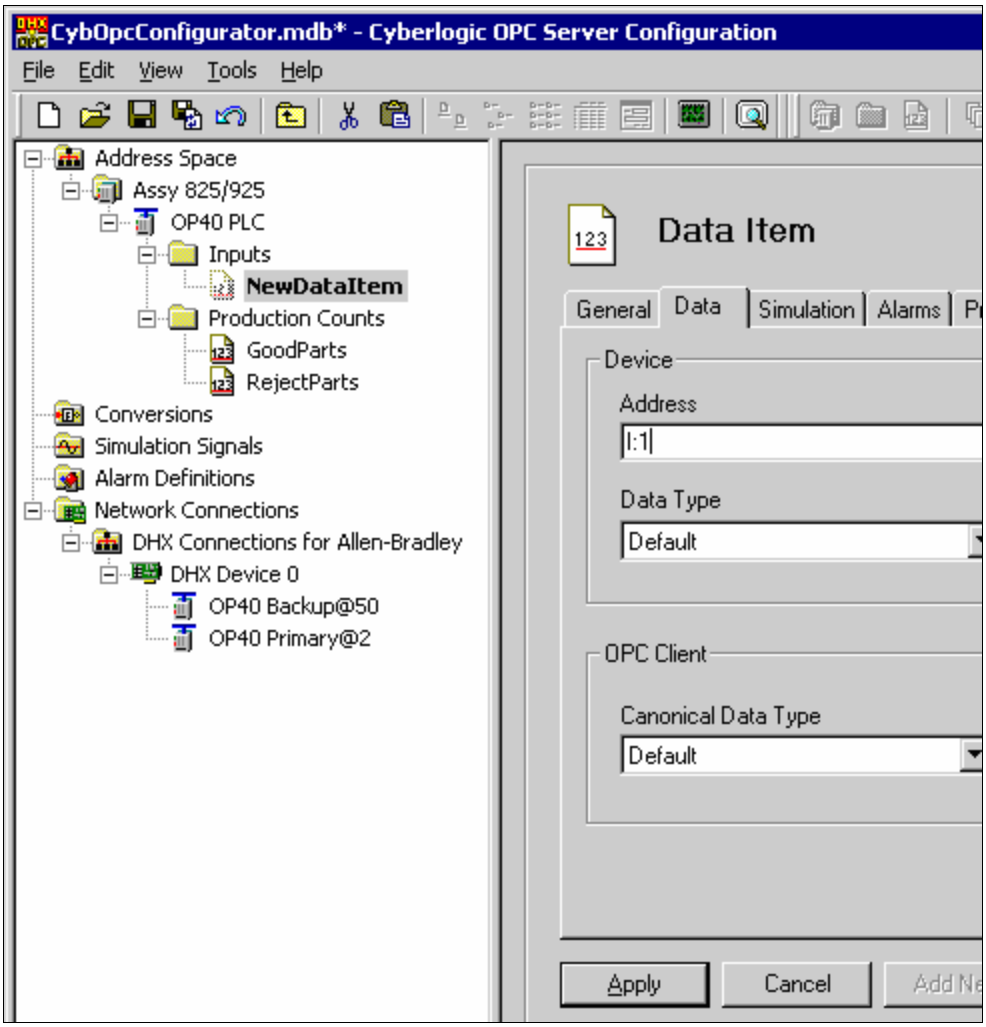

You have manually created two data items that represent some production-related counts. The GoodParts count will be updated using solicited communications, while the RejectParts count will be updated using unsolicited communications. In addition, if the Server does not receive unsolicited updates within one minute, it will downgrade the item's quality from *Good* to *Uncertain*.

Next, you will configure the Data Items for the Inputs folder.

21. Select the *Inputs* folder and select *New/Data Item* from the Edit menu (or right-click on the *Inputs* Folder and select *New/Data Item* from the context menu). Enter a unique name (*I_001*, for this example) and an optional description.

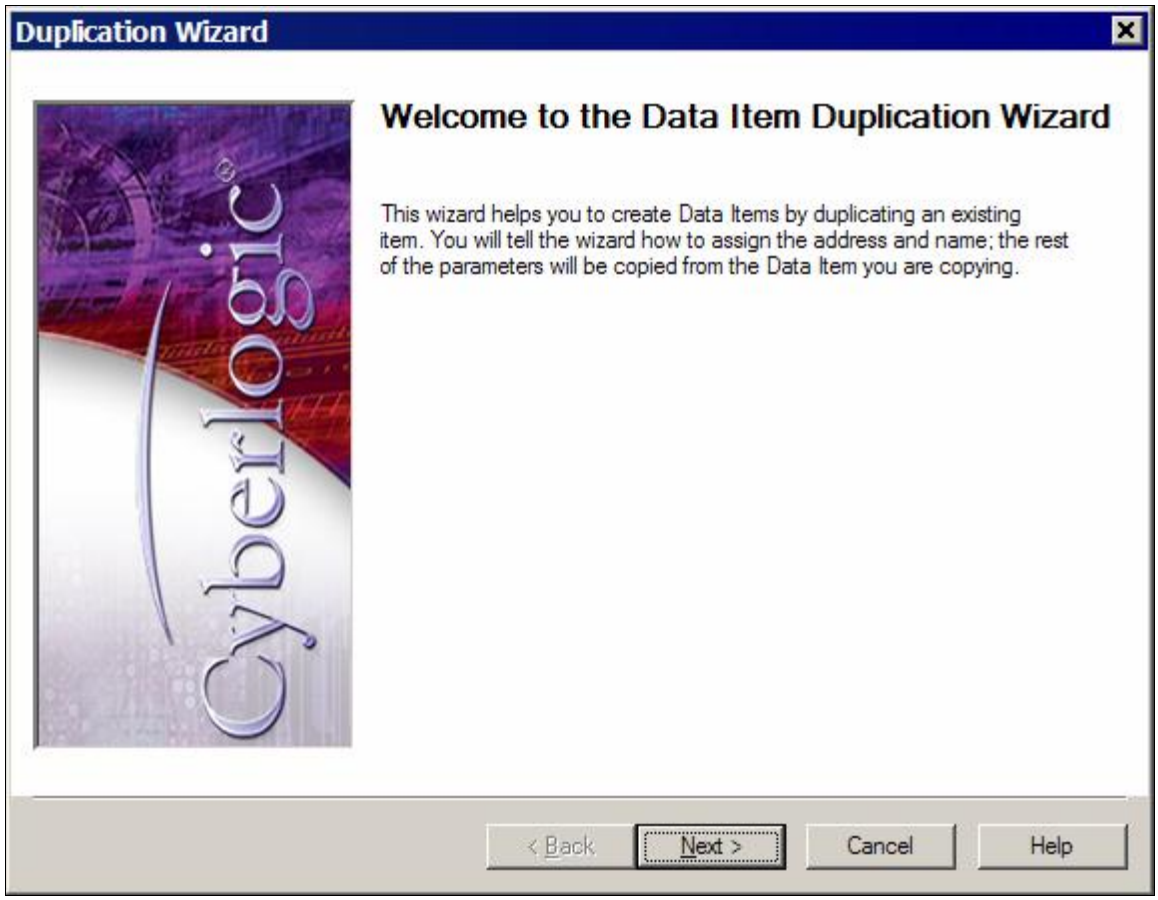


22. Select the *Data* tab, Enter *I:1* in the Address field. Click *Apply* when you are done.

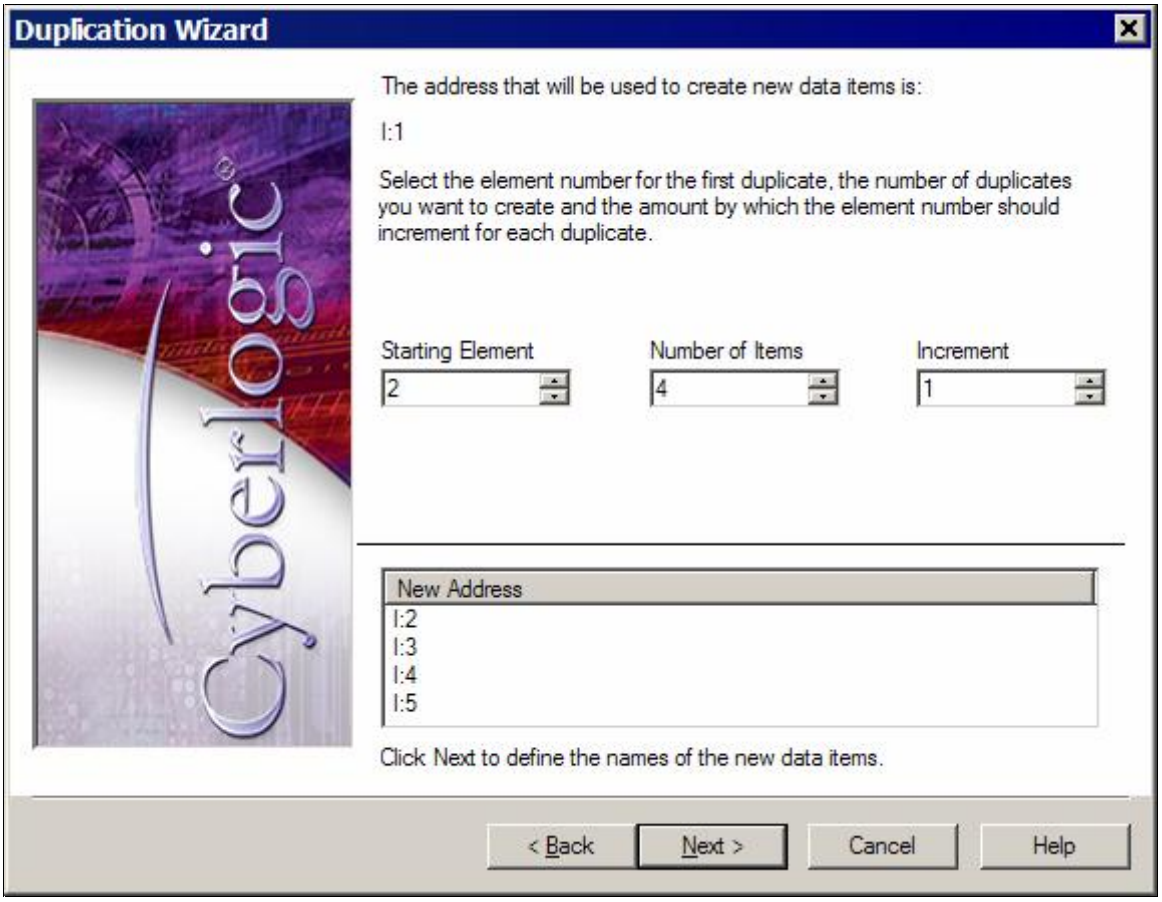



This created a single input. You will now use the Duplication Wizard to create four more input data items in a single operation.

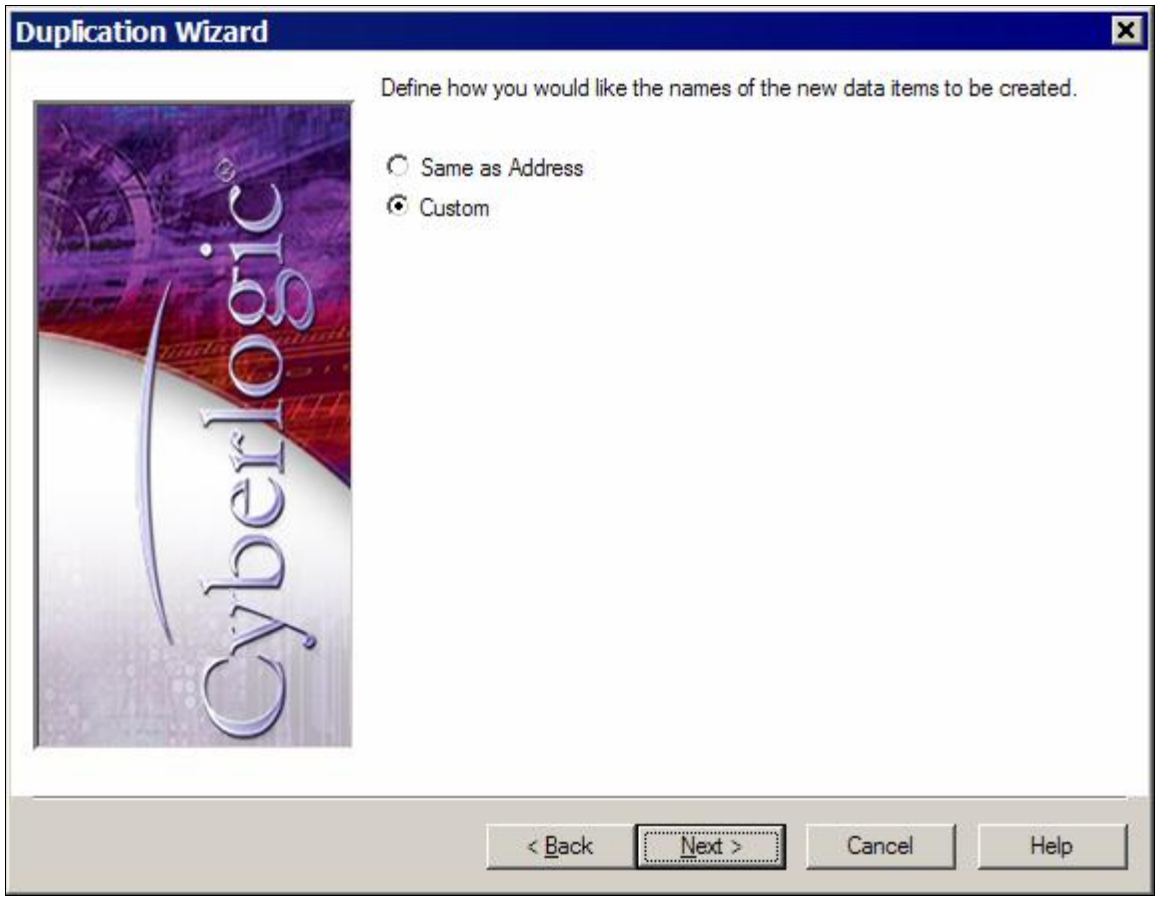
23. Select Data Item I_001. From the Edit menu, select *Duplicate…* (or right-click *I_001* and select *Duplicate* from the context menu). The Duplication Wizard will open. Click *Next* to move to the next screen.



24. On this screen, you will specify the duplicate Data Items you want to create. Notice that the screen tells you which Data Item you are duplicating. Enter *2* for the Starting Element, *4* for the Number of Items and *1* for the Increment. The New Address box shows you the Data Items that the wizard will create. Click *Next* to continue.

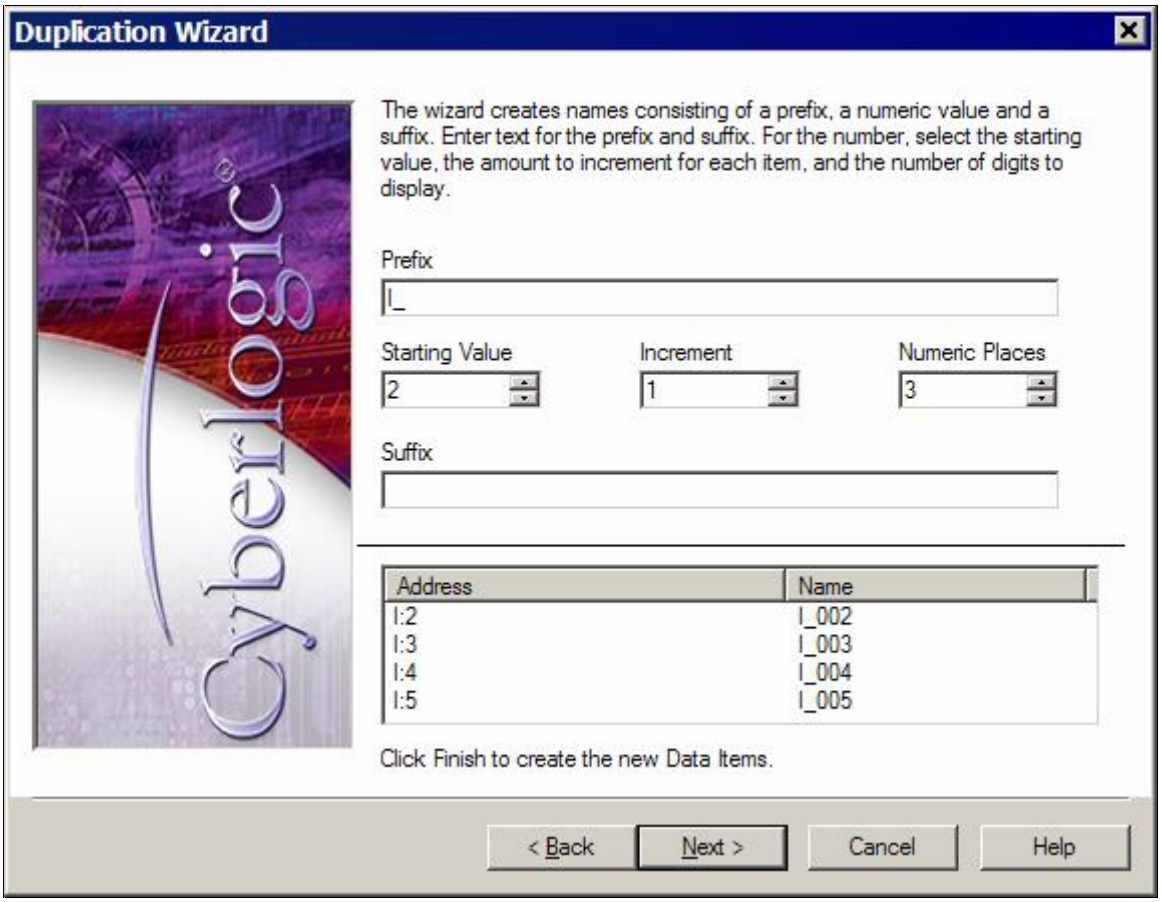


25. You must now decide how you wish to name the Data Items you will create. You may simply use the address as the name or you may create a custom naming scheme. Select *Custom* and click *Next* to continue.

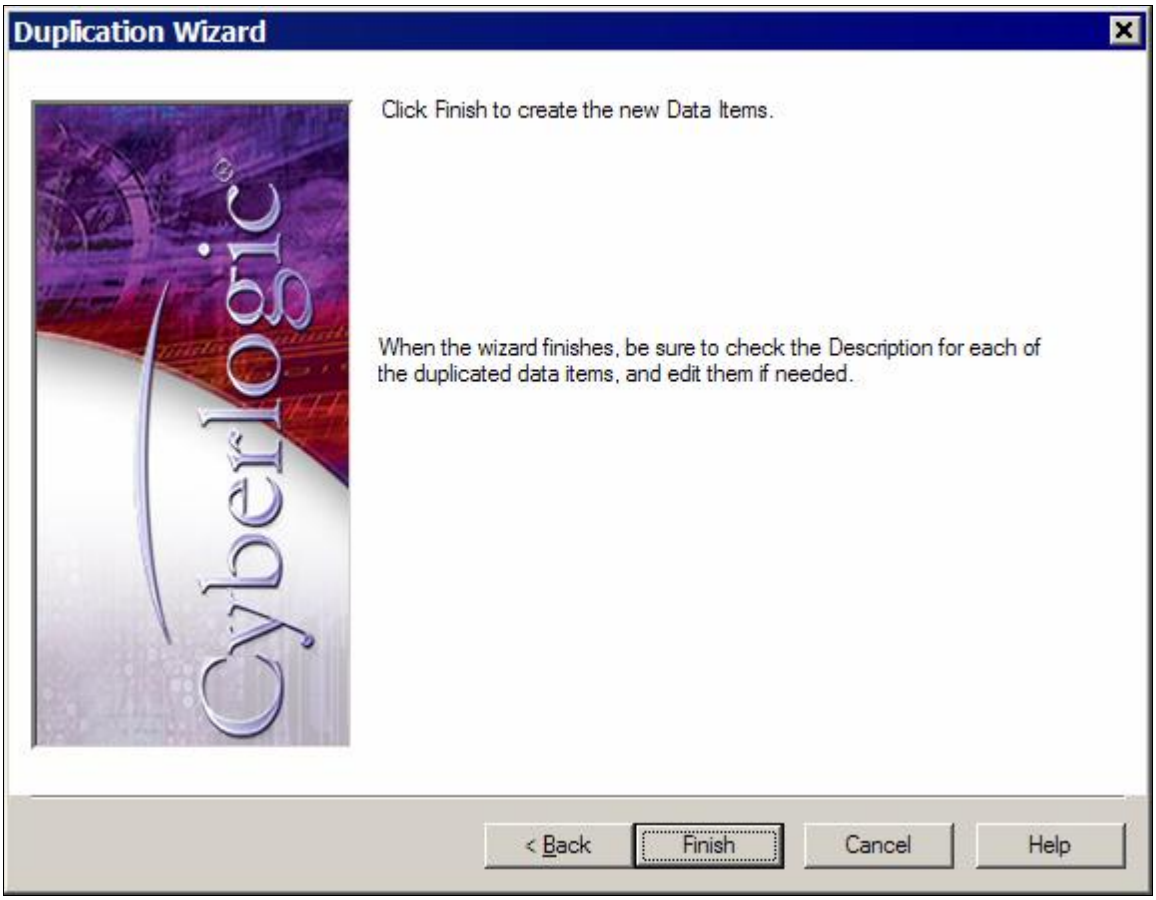


26. The wizard will create names for the Data Items for you. These names will consist of a prefix, a numeric value and a suffix. The first Data Item we created was named *I_001* and we would like the duplicates to have names of the same style. Enter *I_* in the Prefix field. The next three fields define the numeric values to be used. Enter *2* as the Starting Value, *1* for the Increment and *3* for Numeric Places. This causes the duplicates to be consecutively numbered, beginning at 2. It also forces the names to use three digits, inserting leading zeros as needed. No suffix is necessary for this naming scheme, so leave the Suffix field blank. The lower window will show you the names that will be used for each Data Item.

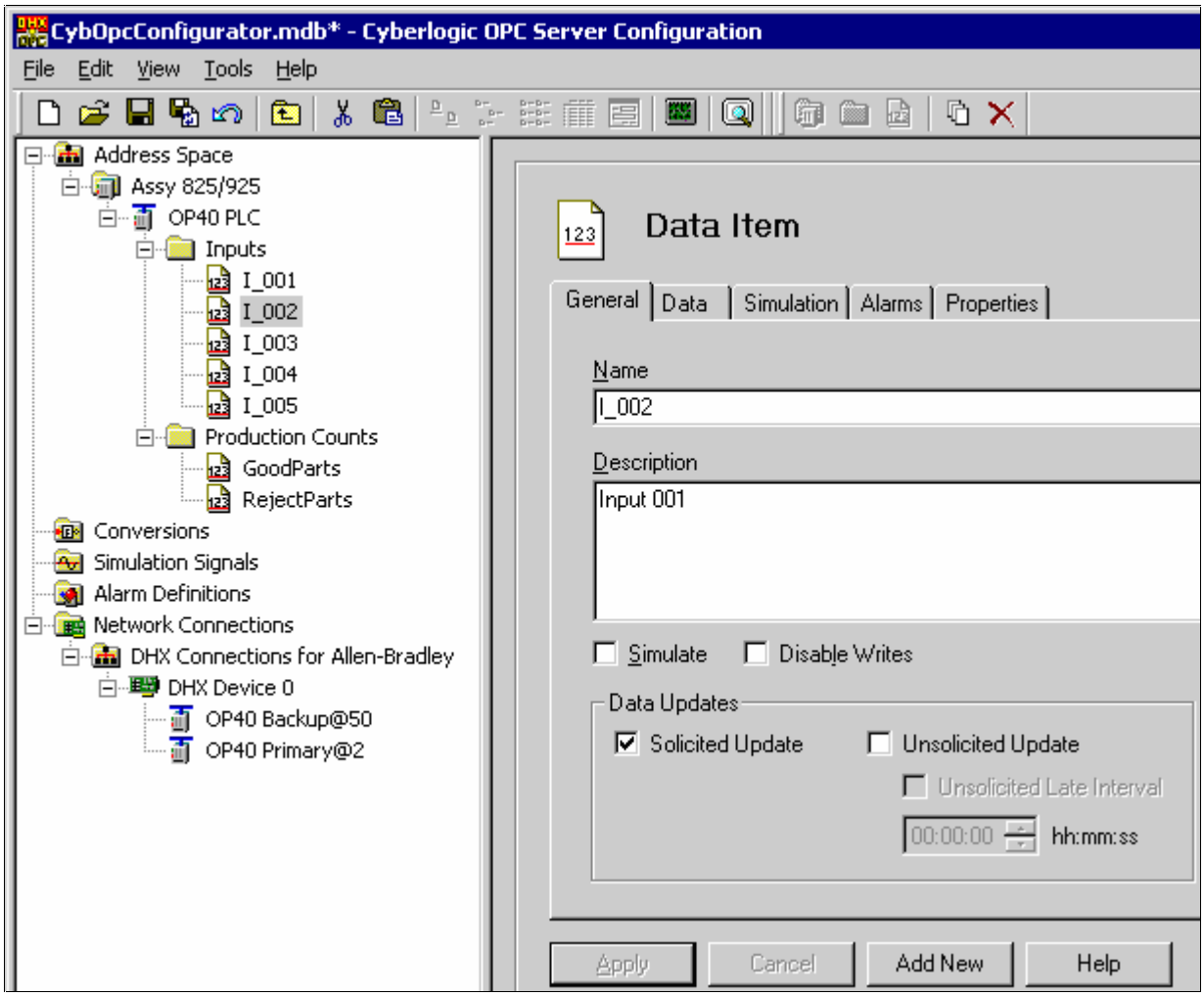
    Click *Next* to continue.



27. Click *Finish* to create the data items and exit the wizard.



28. Your duplicates will be created. Notice, however, that the Description field for each duplicate is the same as the original. You may wish to edit these descriptions.



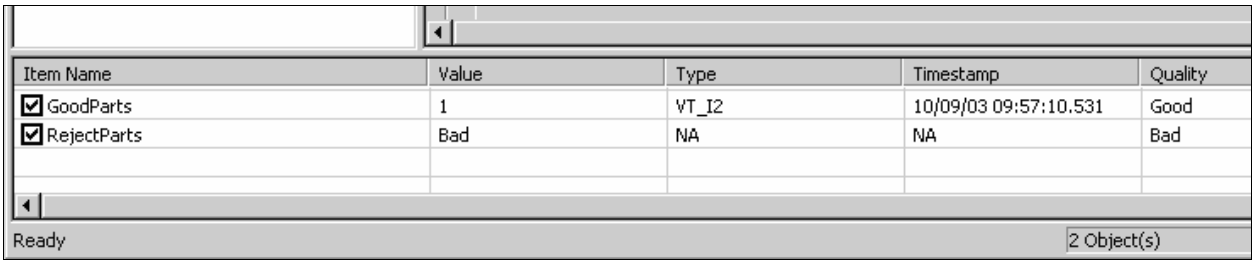

By using the Duplication Wizard, you were able to create four Data Items using just a short, simple procedure. This method is very useful when creating similarly configured Data Items.

29. From the File menu, select *Save & Update Server*.

This completes the typical configuration session of the Cyberlogic OPC Server. The next step will introduce you to the diagnostic features of the product.

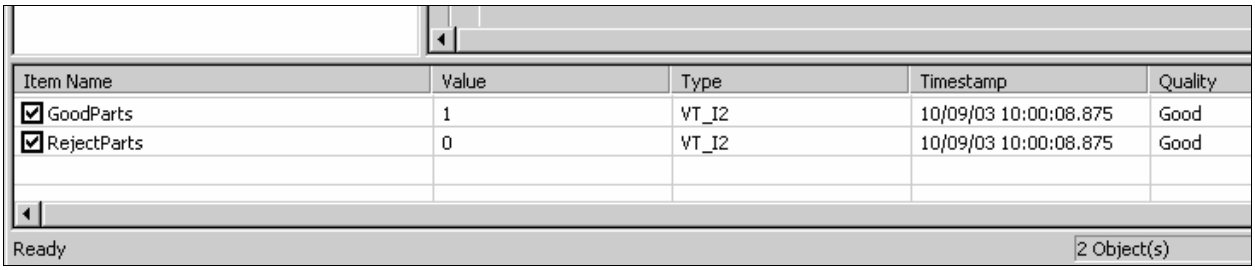
The Cyberlogic OPC Server Configuration Editor includes a built-in utility called the Data Monitor. This diagnostic tool allows you to view the status and values for data items in the currently selected Folder. To enable the Data Monitor, select *Data Monitor* from the View menu or right-click the desired Folder and select *Data Monitor* from the context menu.

30. Select the *Production Counts* folder and select *Data Monitor* from the View menu (or right-click on the *Production Counts* folder and select *Data Monitor* from the context menu). Check the *Enable* boxes to the left of each data item in the Data Monitor view window.




31. Because the GoodParts data item was configured for solicited update, the Server quickly retrieved new values with Good quality. However, the RejectParts data item will show Bad quality until the primary PLC sends an update.

32. Program the primary PLC to send data to the N7:1 register address in the Server.




33. Now, the RejectParts shows data with Good quality.

34. Disconnect the Data Highway Plus cable from the primary PLC and wait for one minute. Because the PLC will not be able to send additional unsolicited updates, the quality of RejectParts will change from Good to Uncertain.

35. With the primary PLC unable to communicate, the server will automatically switch to the backup PLC. Notice that the quality of the GoodParts item still shows Good quality.

This concludes the tutorial. To learn more about the advanced features of the server, refer to the Theory of Operation section.

# Creating Network Connections and Nodes

This section will take you through the steps required to set up new Network Connections and Network Nodes. You would do this if you added a new network or controller to the system.

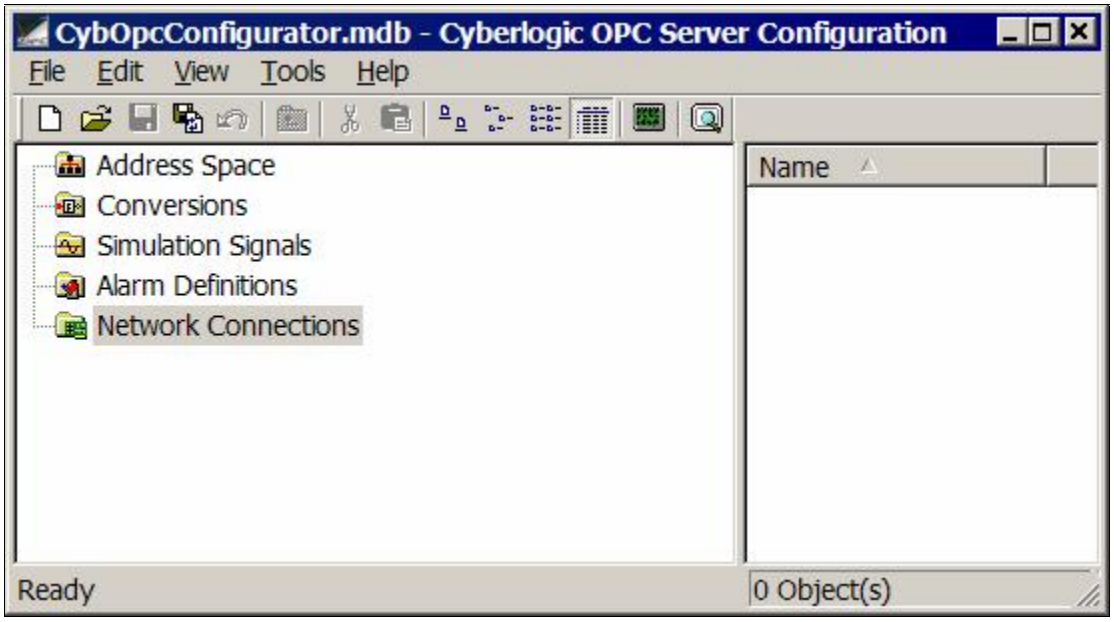
If you need a quick-start guide or a step-by-step configuration session tutorial, go to the Typical Configuration Session section. For detailed information on the features of the driver agent's Configuration Editor, refer to the help file for the driver agent you are using.

For this example, we will assume you are using a DHX driver agent. The steps for other driver agents will be similar. The first step in configuring the Cyberlogic OPC Server is to create at least one DHX device. If you wish to do the configuration on-line so that you can take advantage of the automatic configuration feature for Network Connections and Network Nodes, you must have a driver installed. Depending upon your communication network, you would need at least one of the following Cyberlogic driver products:

- DHX Driver
- Ethernet DHX Driver
- Serial DHX Driver
- DHX Gateway Driver

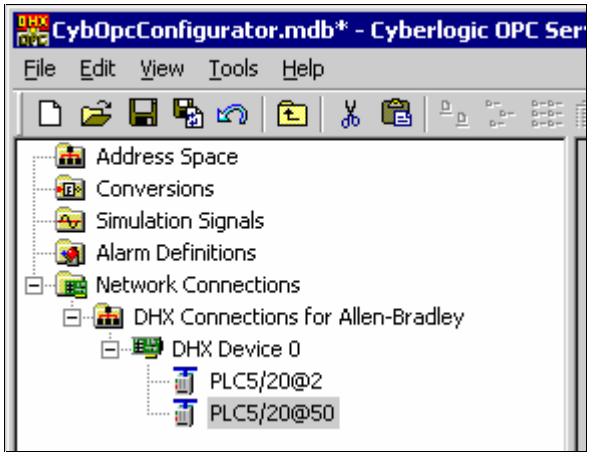
Refer to the driver-specific help file for information on configuring the DHX devices. Use the DHX Demo program to verify proper operation of the installed drivers. After you have configured and tested the DHX devices, follow these steps:

1. First, open the Windows Start menu, locate the *DHX OPC Server* submenu and select the *OPC Server Configuration* menu item.




2. Since you are running the Cyberlogic OPC Server Configuration Editor for the first time, the editor will prompt you to go to the File menu and open a new configuration database. You will start with an empty database.

3. Select the *Network Connections* root folder and select *Auto Config* from the Edit menu (or right-click on the *Network Connections* root folder and select *Auto Config* from the context menu). The editor will try to find all Network Connections and automatically detect and configure all Network Nodes. When it is done, you will see a configuration similar to the following in the Network Connections tree:




4. This screen shows that the editor has detected two programmable controllers, one at node address 2 (PLC-5/20@2) and the other one at the node address 50 (PLC-5/20@50).

| **Note:** | ControlLogix nodes do not report enough information to permit Auto Config to identify them. They will be detected, but will be reported simply as DHX nodes of unknown type. |
|---|---|

5. If you wish to configure all or part of the Network Connections manually, refer to the Manual Configuration section, below, for instructions on how to create the Network Connections and to the specific driver agent documentation for instructions on creating Network Nodes.

6. From the File menu, select *Save & Update Server*.

7. You have now completed the minimum required configuration for the Server. You need not do any additional configuration if you limit yourself to DirectAccess of the PLC registers.

Next, you should configure the following items if they are required.

- Alarm Definitions
- Conversions
- Simulation Signals

After the above steps are completed, most of your configuration will be concentrated on the Address Space tree. Refer to the documentation for the driver agent you are using for details on how to configure the Address Space.

# OPC Server Configuration Editor

This section gives you detailed information on how to modify an existing Server configuration.

For information on setting up Network Connections and Network Nodes, refer to the Creating Network Connections and Nodes section. If you need a quick-start guide or a step-by-step configuration session

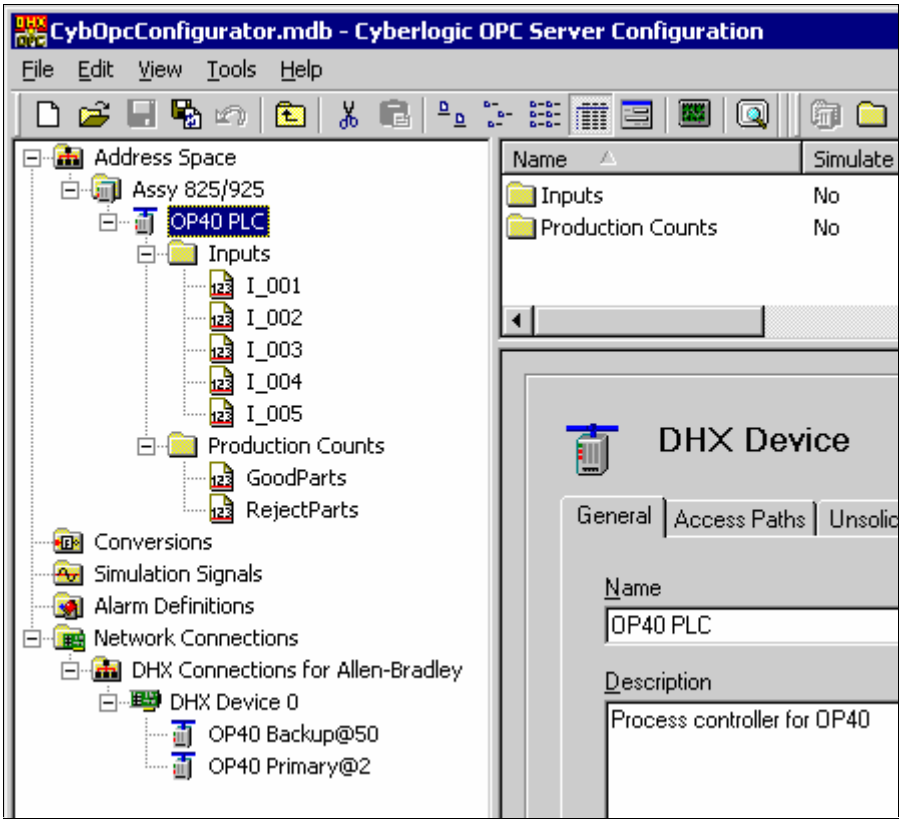
tutorial, go to the Typical Configuration Session. To learn more about the advanced features of the Server, refer to the Theory of Operation section.

To start the editor, open the Windows Start menu, locate the DHX OPC Server submenu (or the submenu for the OPC Server you are working with) and select the *OPC Server Configuration* menu item.

The Cyberlogic OPC Server Configuration Editor allows the user to create and modify the configuration file used by the runtime module. It is needed only to generate configuration files and is not otherwise required for the operation of the runtime module.

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

An Explorer-like user interface allows easy manipulation of the configuration file and provides numerous common Windows functions, such as copy and paste, drag and drop and context-based pop-up menus.




The left pane of the main workspace window includes the five main configuration trees:

- Address Space Tree
- Conversions
- Simulation Signals
- Alarm Definitions
- Network Connections Tree

The following sections provide complete descriptions of these trees.

# *Network Connections Tree*

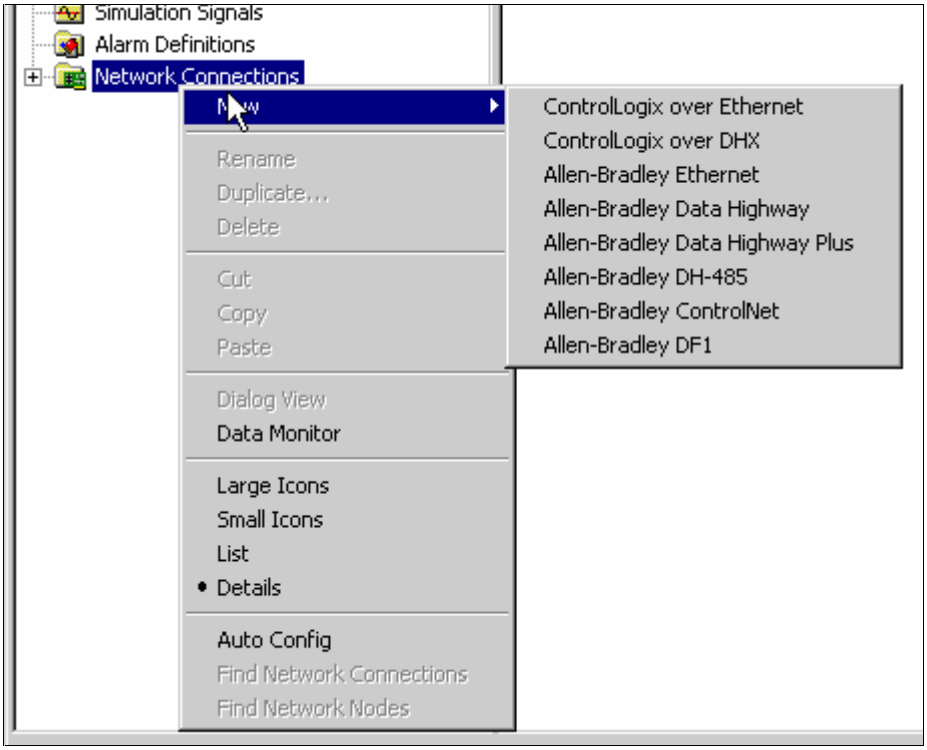
The features and details of the Network Connections will vary depending upon the driver agent you are using. Each driver agent has its own help file. To access that help file, select the driver agent folder (or any folder below the driver agent folder) for which you need help. The driver agent folders are those folders directly below the Network Connections folder. A *Help* button will be available in the right pane of the Editor window.

If the driver agent folder you need is not shown, you can create a Network Connection of that type, and then you will be able to access the help. Creating a Network Connection will create the driver agent folder. There are two ways to do this, Manual and Automatic.

## Manual Configuration

You may prefer to configure your Network Connections and Network Nodes manually. This will be necessary if you are doing the configuration on a computer that is not connected to the target networks or if you wish to change the default values selected during an Auto Configuration.

Select the *Network Connections* root folder, open the *Edit* menu item and select *New*, and then select the desired network type (or right-click on the *Network Connections* root folder and select *New*, then the network type from the context menu). The Editor will create the proper driver agent and Network Connection folders.




| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## Auto Configuration

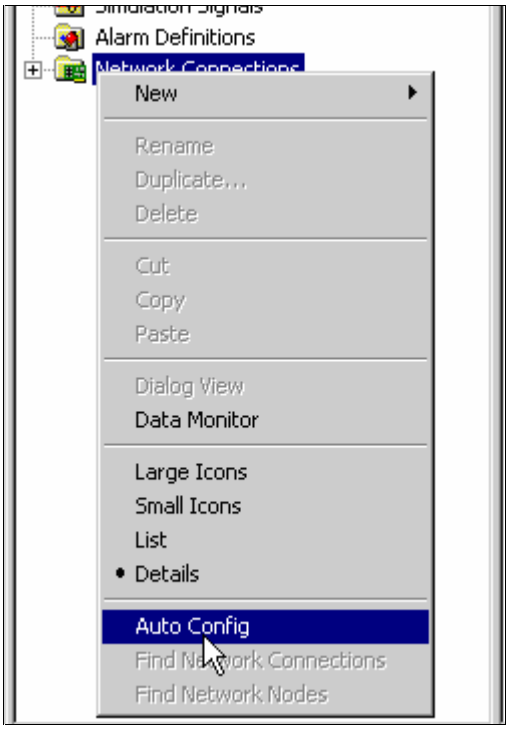
The simplest method of configuration is Auto Configuration.

| | |
|---|---|
| **Caution:** | Before you can use the Auto Configuration feature, you must install and configure the low-level device drivers that the OPC Server will use. The configuration editors supplied with the device drivers will allow you to create the devices that the OPC Server will then be able to detect. |

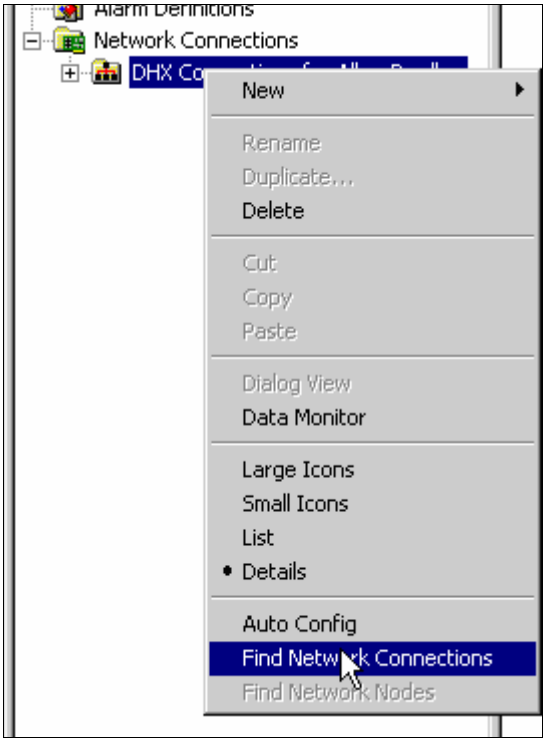
| | |
|---|---|
| **Note:** | Some types of nodes, such as ControlLogix, do not report enough information to permit Auto Config to identify them. They may be detected, but will be reported simply as nodes of unknown type. |

To begin the Auto Configuration process, select the *Network Connections* root folder.

To find all Network Connections and detect and configure all Network Nodes, select *Auto Config* from the Edit menu (or right-click on the *Network Connections* root folder and select *Auto Config* from the context menu).



After a driver agent folder has been created, you can automatically find all Network Connections of that type available on your system. To do this, select the driver agent folder, then open the *Edit* menu and select the *Find Network Connections* (or right-click the *Network Connections* root folder and select *Find Network Connections* from the context menu). Typically, you would do this if you did part of the configuration while not connected to the target network and want to quickly finish the configuration once you are connected.




| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

# *Address Space Tree*

The Address Space Tree describes the hierarchical address structure of the Cyberlogic OPC Server. It consists of branches (Device Folders, Devices and Folders) and leaves (Data Items). The root folder of the Address Space tree may contain Device Folders and Devices. The intent of this structure is to permit the user to organize the Data Items into logical groups.

The features and details of the Devices in the Address Space tree will vary depending upon the driver agent you are using. Each driver agent has its own help file. To access that help file, select an Address Space Device of the driver agent for which you need help. A *Help* button will be available in the right pane of the Editor window.

If the Device type you need is not shown, simply create a Device of that type and you will be able to access the help. Select either the Address Space root folder or an existing Device Folder. From the Edit menu, select *New/Device* (or right-click on the folder and select *New/Device* from the context menu). Select the desired driver agent to create the Device. Depending upon the selected driver agent, the

required information that needs to be entered in the right pane will vary (Click the *Help* button for more information). Enter the required information and then click *Apply* when you are done.

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

# *Conversions*

The raw data associated with a Data Item may be a process value from an instrument. In most cases, these measurements are not expressed in engineering units. To simplify operations on the data, the Cyberlogic OPC Server allows you to associate a Conversion with each Data Item. A user can define many different types of conversions. A number of Data Items can then use each such Conversion. As a result, the user need not define the same Conversion many times over.

The Conversions feature also supports data range clamping. You can instruct the Server to clamp the data within a specified range of engineering unit values. The clamping feature is available even if you choose not to apply a linear or square root conversion function to the data.

## Linear and Square Root Functions

Two types of conversion functions are available, linear and square root. These will handle the conversion needs for most instrument types.

| | |
|---|---|
| **Caution:** | The conversion functions work on the data as it is passed in both directions. You will specify the conversion to be applied to data passed from the server to the client application. When data is passed from the client to the server, the inverse conversion will be applied. This means that the client must accept data as being in engineering units and write data in engineering units. |

In this section, we will discuss the functions used to make these conversions, as an aid in helping you to apply them to your application. The definitions of the variables used are as follows:

$CV$ is the Converted Value reported to the client application.

$DV$ is the raw Data Value received from the instrument.

$ID_L$ and $ID_H$ are the low and high Instrument Data values you will specify during configuration.

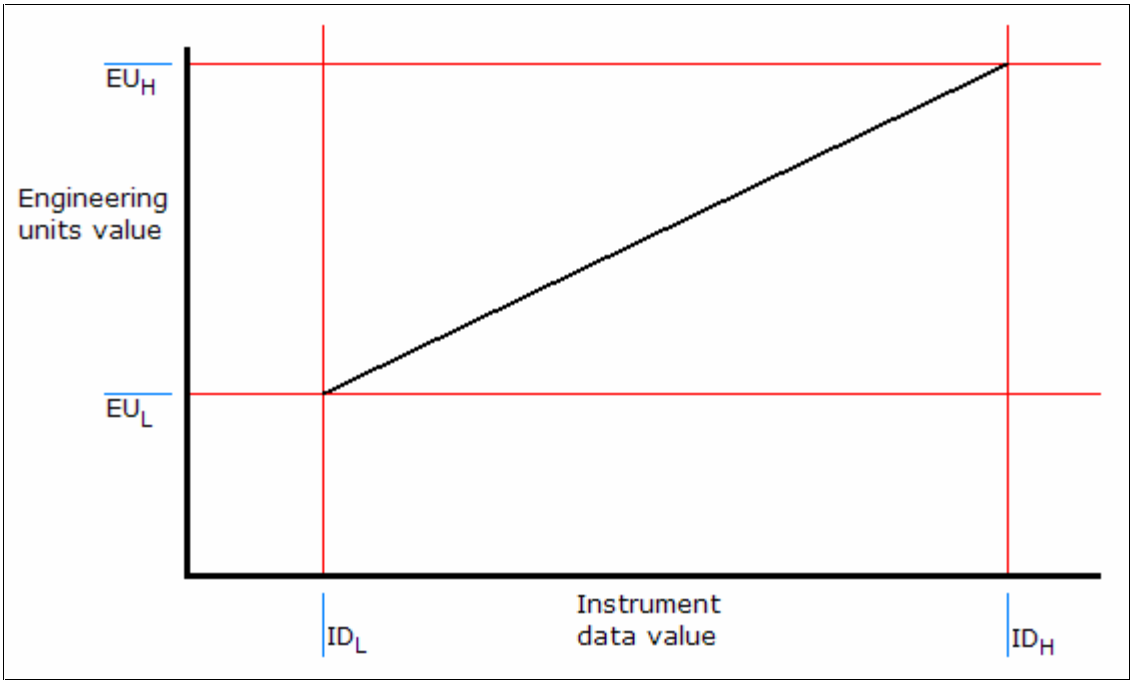
$EU_L$ and $EU_H$ are the low and high Engineering Units values you will specify during configuration.

### *Linear Conversion*

This performs a standard, straight-line linear conversion with offset, according to the following formula:

$$CV = \left( \frac{DV - ID_L}{ID_H - ID_L} \right) \bullet \left( EU_H - EU_L \right) + EU_L$$

The resulting function looks like this:

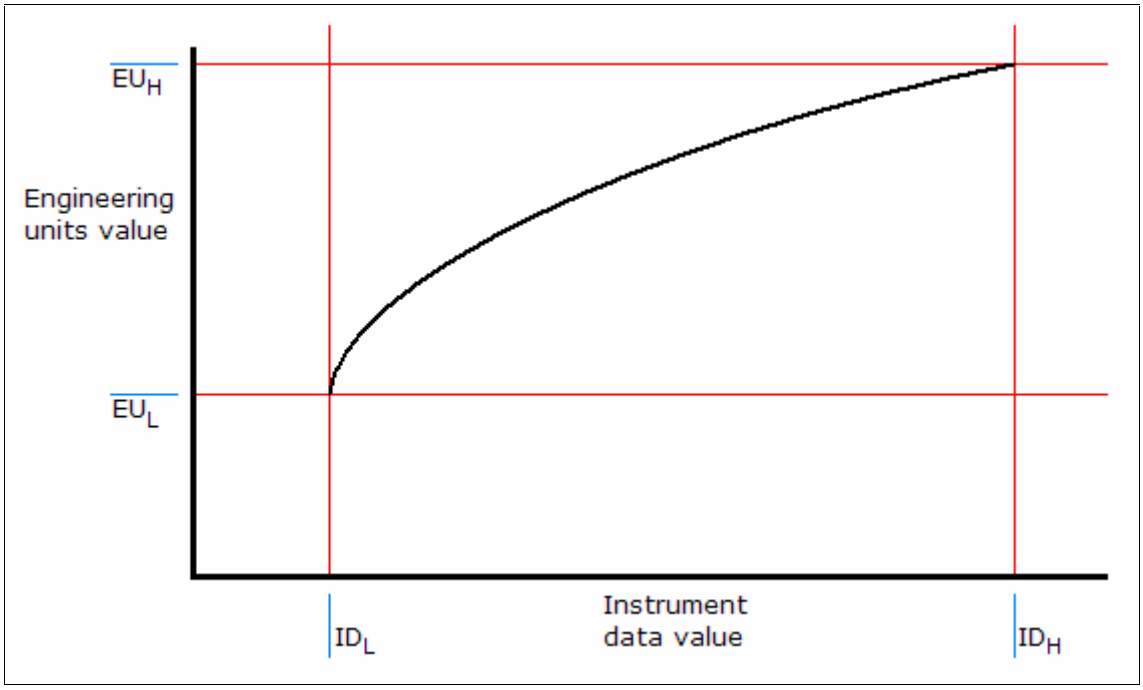


## Square Root Conversion

This is similar to the linear conversion, but is proportional to the square root of the value. The formula is:

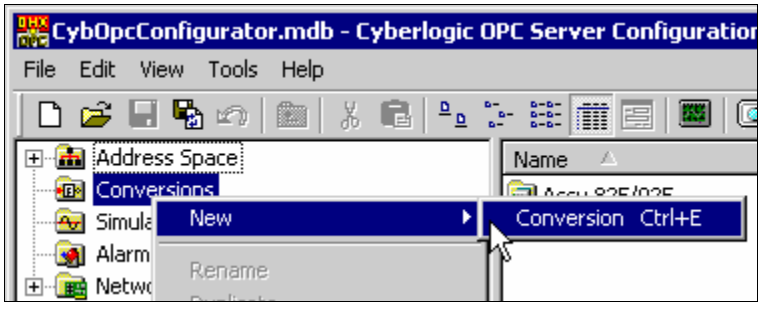
$$CV = \sqrt{\frac{DV - ID_L}{ID_H - ID_L}} \bullet \left(EU_H - EU_L\right) + EU_L$$

The resulting function looks like this:




## Creating a New Conversion

Select the *Conversions* root folder and select *New/Conversion* from the Edit menu (or right-click on the *Conversions* root folder and select *New/Conversion* from the context menu). Enter the information required in the Conversion dialog box, as described below. Click *Apply* when you are done.

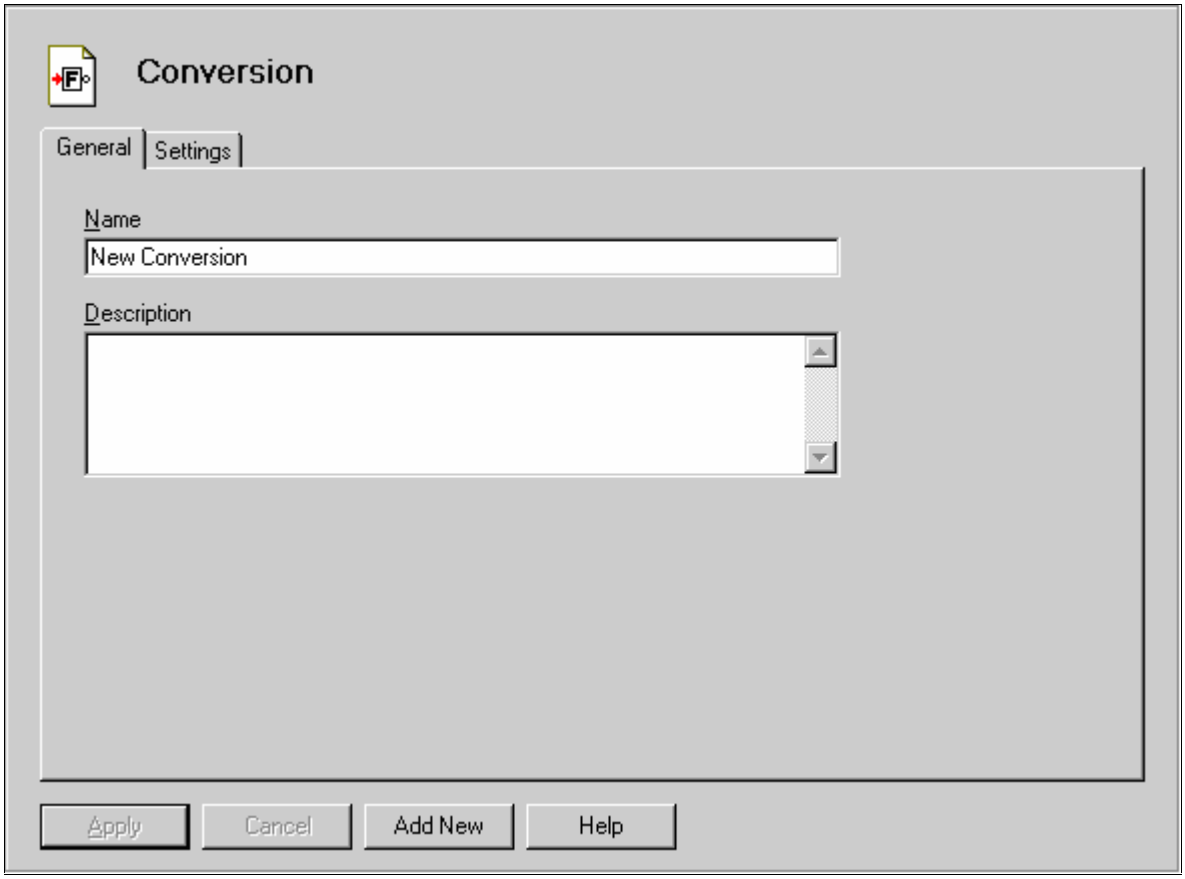


## Deleting a Conversion

Select an existing conversion and press the *Delete* key (or right-click on the conversion and select *Delete* from the context menu).

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## General Tab




### *Name*

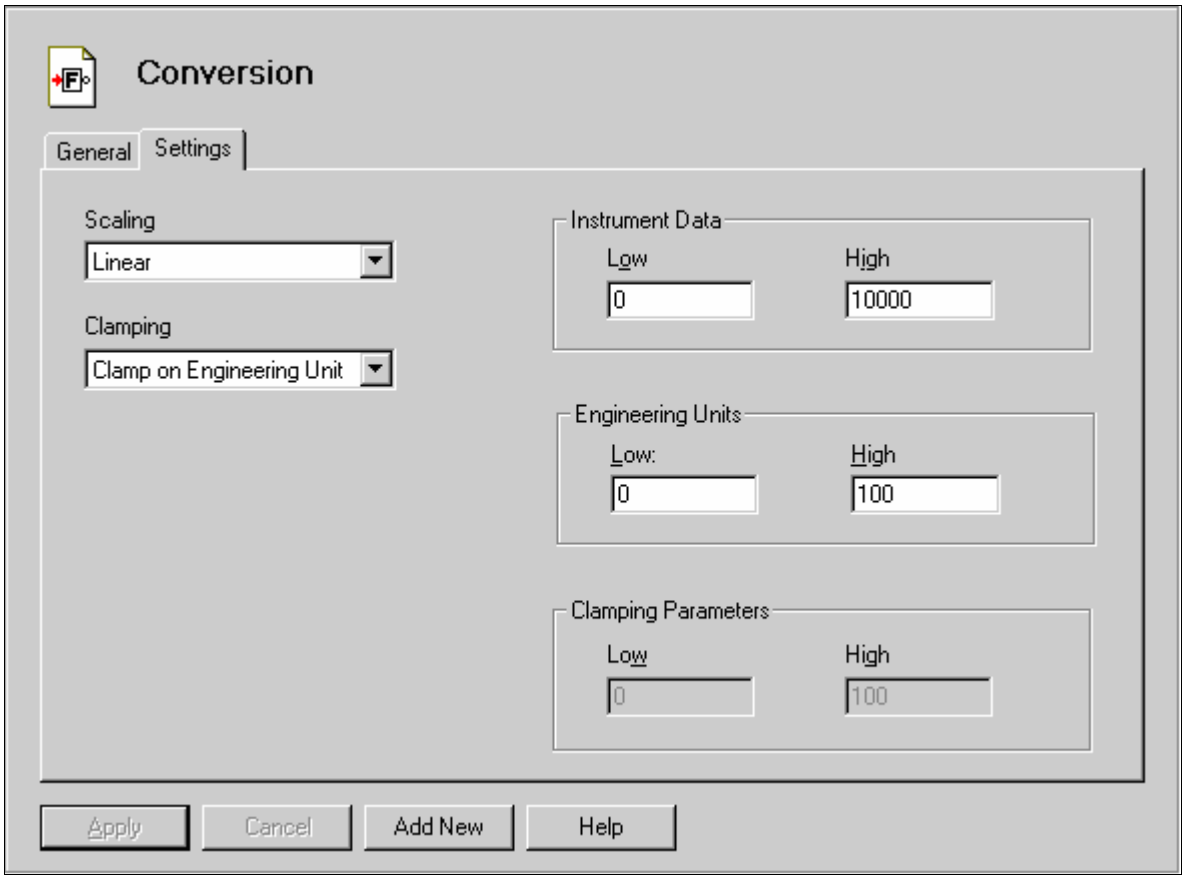
The name identifies the Conversion. It can be up to 50 characters long, may contain spaces, but must not begin with a space. It also must not contain any periods.

### *Description*

This optional field further describes the conversion. It can be up to 255 characters long.

## Settings Tab

All of the values you will specify on this tab are taken as 64-bit floating point numbers. Consequently, they may be positive or negative and have magnitudes in the range of $4.9 \times 10^{-307}$ to $1.8 \times 10^{+308}$ or 0.




### *Scaling*

Select the type of scaling you wish to use. The choices are *None, Linear* and *Square Root*. If you select linear or square root scaling, you must enter values for the Instrument Data and Engineering Units. These values specify the scaling factors applied to the data.

### *Instrument Data*

Enter the limits of the raw value from the instrument. The Instrument Data value entry fields are available only for Linear and Square-Root conversions.

### *Engineering Units*

Enter the limits of the scaled value. You may enter Engineering Units values even if you choose *None* for the conversion type. This allows you to clamp on the Engineering Units limits, if you wish.

### *Clamping*

Select the type of clamping you wish to use. The choices are *None, Clamp on Engineering Unit* and *As Specified*.

If you choose *Clamp on Engineering Unit*, the value will be clamped within the range specified in the *Engineering Units* box. If you choose *As Specified*, you must specify the desired clamping limits in the *Clamping Parameters* box.

When clamping is used, any values below the minimum of the clamping range will be set to the minimum value and any values above the maximum will be set to the maximum value. Clamping is applied to the data after the scaling conversion. If you choose not to scale the data, you may still use the clamping feature. The limits, whether *Engineering Units* or *As Specified*, would then be applied to the raw data.

### *Clamping Parameters*

Enter the values for the clamping range. These fields are available only if you choose *As Specified* for the clamping type.

## *Simulation Signals*

The Server can simulate the data for each Data Item according to a predefined formula. This makes it easy to perform client-side testing without the need for a physical device,

A user can define many different types of Simulation Signals. A number of Data Items can then use each such signal. As a result, the user need not define the same Simulation Signal many times over.

The Server can generate the following types of Simulation Signals:

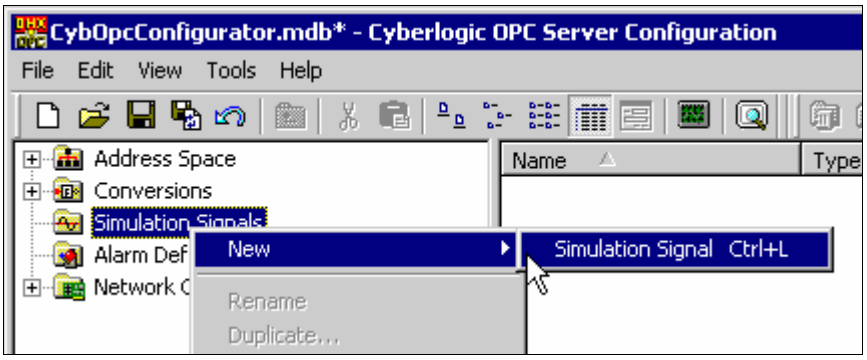
- Read Count
- Write Count
- Random
- Ramp
- Sine
- Square
- Triangle
- Step

Most of these signals have parameters that define properties such as amplitude, phase and number of steps.

You can enable data simulation at any level in the Server Address Space. Enabling data simulation at any level automatically enables it at all levels below. This permits quick switching between simulated and real data for a large number of Data Items.

## Creating a New Simulation Signal

Select the *Simulation Signals* root folder and select *New/Simulation Signal* from the Edit menu (or right-click on the *Simulation Signals* root folder and select *New/Simulation Signal* from the context menu). Enter the information required in the Simulation Signal dialog box, as described below. Click *Apply* when you are done.

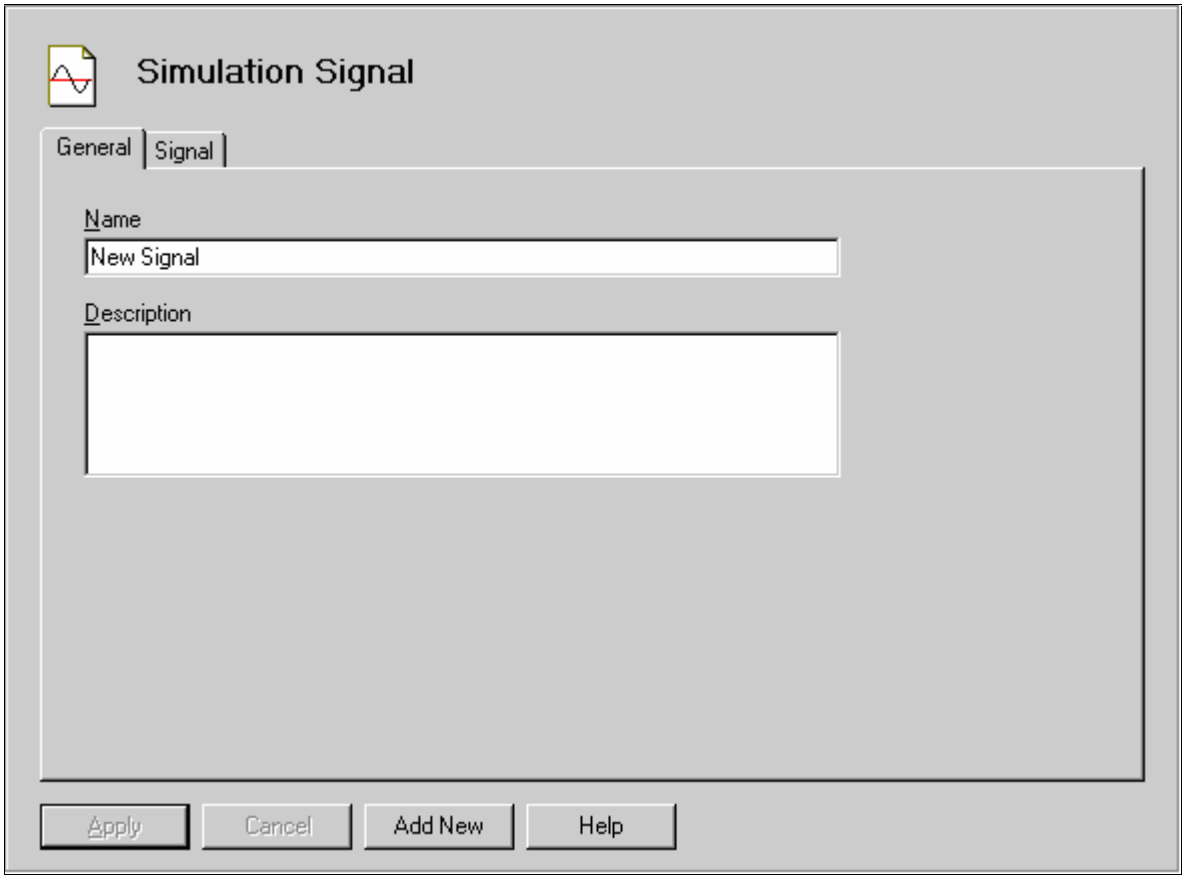



## Deleting a Simulation Signal

Select an existing Simulation Signal and press the *Delete* key (or right-click on the Simulation Signal and select *Delete* from the context menu).

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## General Tab



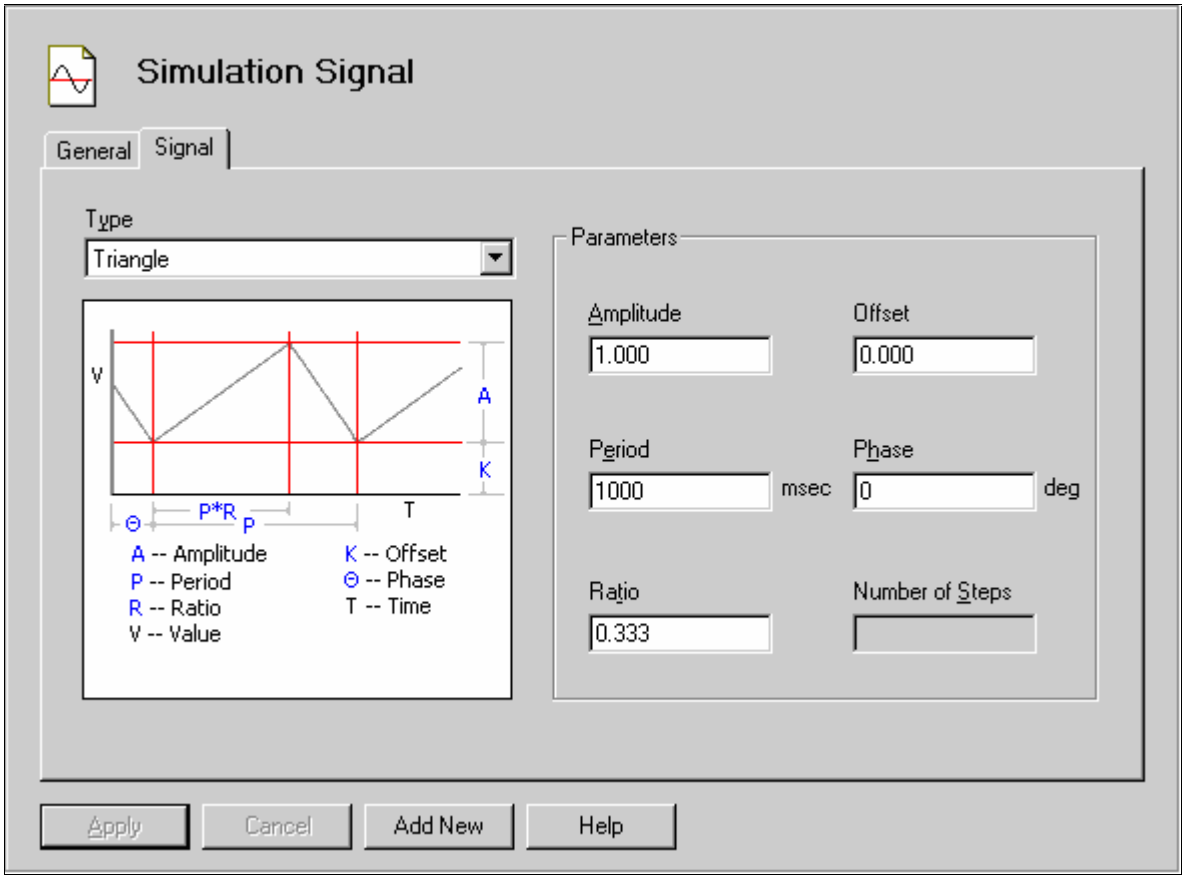

### *Name*

The name identifies the Simulation Signal. It can be up to 50 characters long, may contain spaces, but must not begin with a space. It also must not contain any periods.

### *Description*

This optional field further describes the Simulation Signal. It can be up to 255 characters long.

## Signal Tab




### *Type*

Select the Simulation Signal type from the drop-down box. The available signal types are:

- Write Count: Each write request increments the count.
- Read Count: Each read request increments the count.
- Random
- Ramp
- Sine
- Square
- Triangle
- Step

The signal type selection will enable the needed parameter fields, if any. You must enter the values for the parameter fields that are not dimmed.

### *Amplitude*

This is the peak value of the signal, measured from the Offset level. All signal types except Write Count and Read Count use the Amplitude parameter.

### *Offset*

This is a fixed offset for the value of the Data Item. The generated waveform varies around this value. All signal types except Write Count and Read Count use the Offset parameter.

### *Period*

This is the period of one cycle of the waveform, specified in milliseconds. The Ramp, Sine, Square, Triangle and Step functions use this parameter.

### *Phase*

This is a phase offset for the waveform, specified in degrees. The value you specify will be taken as a leading phase shift and cannot be a negative number. To create a lagging phase shift, subtract the desired shift from 360. For example, if you want a leading shift of 20 degrees, you would enter 20. If you want a lagging shift of 20 degrees, you would enter 340. The Ramp, Sine, Square, Triangle and Step functions use this parameter.

### *Ratio*

The Ratio is a decimal fraction between 0 and 1. Only the Square and Triangle functions use this parameter.

For the Square function, it specifies the fraction of the cycle during which the value is at the low level. A Square function with a Period of 1000 ms and a Ratio of 0.4 would be low for 400 ms and high for 600 ms.

For the Triangle function, it specifies the fraction of the cycle during which the signal is rising. A Triangle function with a Period of 1000 ms and a Ratio of 0.4 would rise for 400 ms and fall for 600 ms.

### *Number Of Steps*

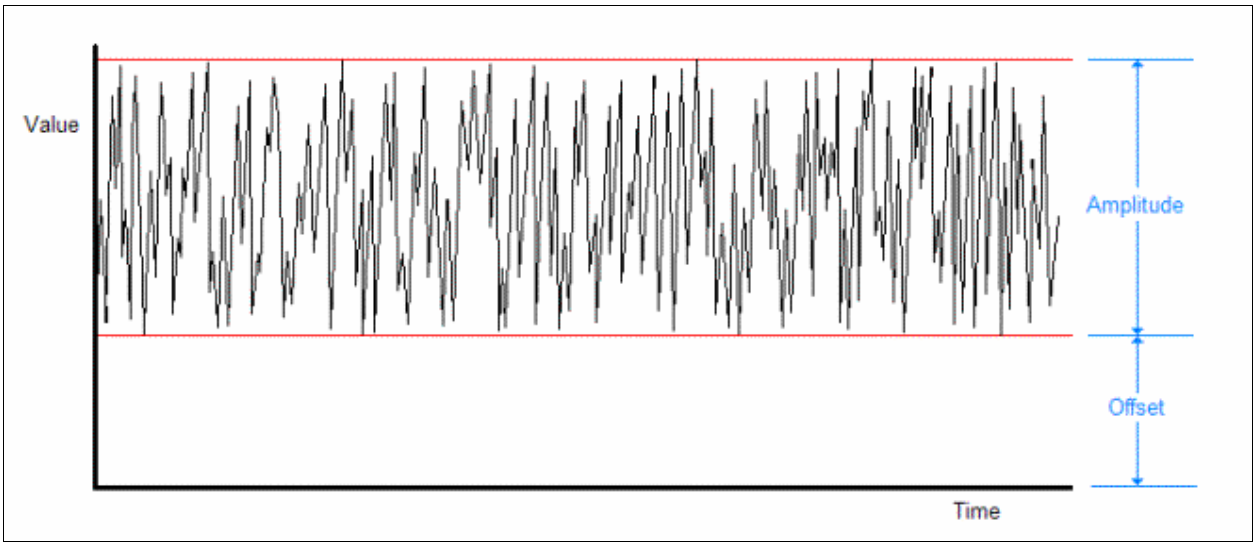
Only the Step function uses this parameter. It specifies the number of value levels in each signal period. The steps are of equal height and width.

## Waveform Illustrations

Here is a set of figures showing the waveforms available for the Simulation Signals. These will assist you in understanding the parameters and specifying the waveform you want.

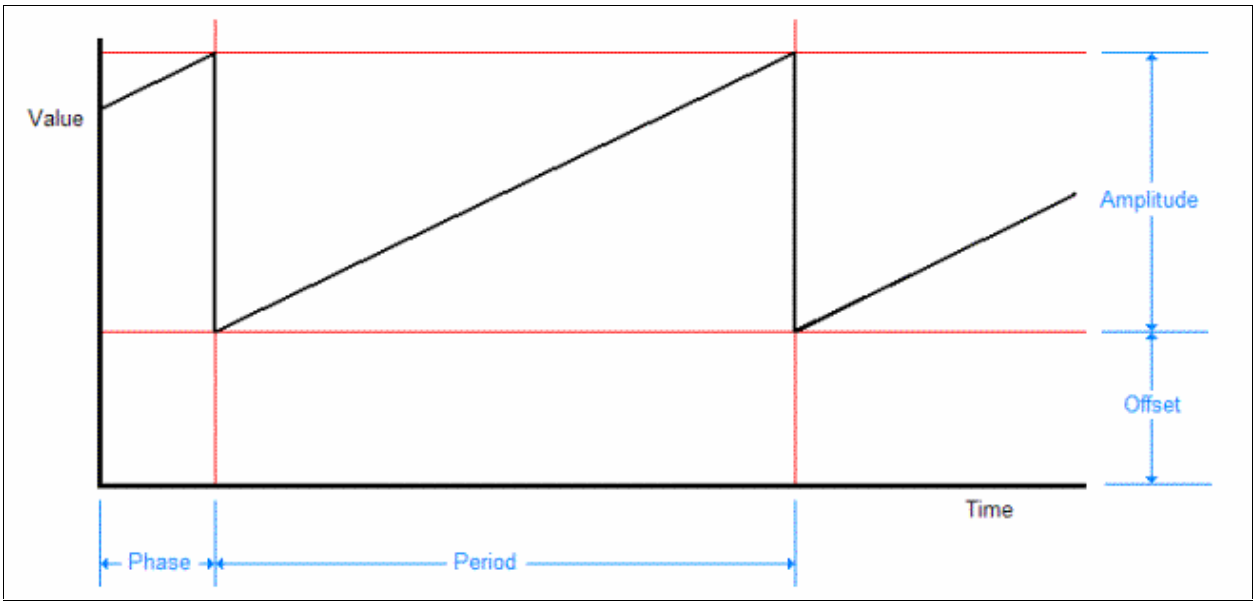
### *Random*

With this function, as you would expect, the value varies randomly. The value of Offset sets the minimum and the Amplitude – added to Offset – sets the maximum.



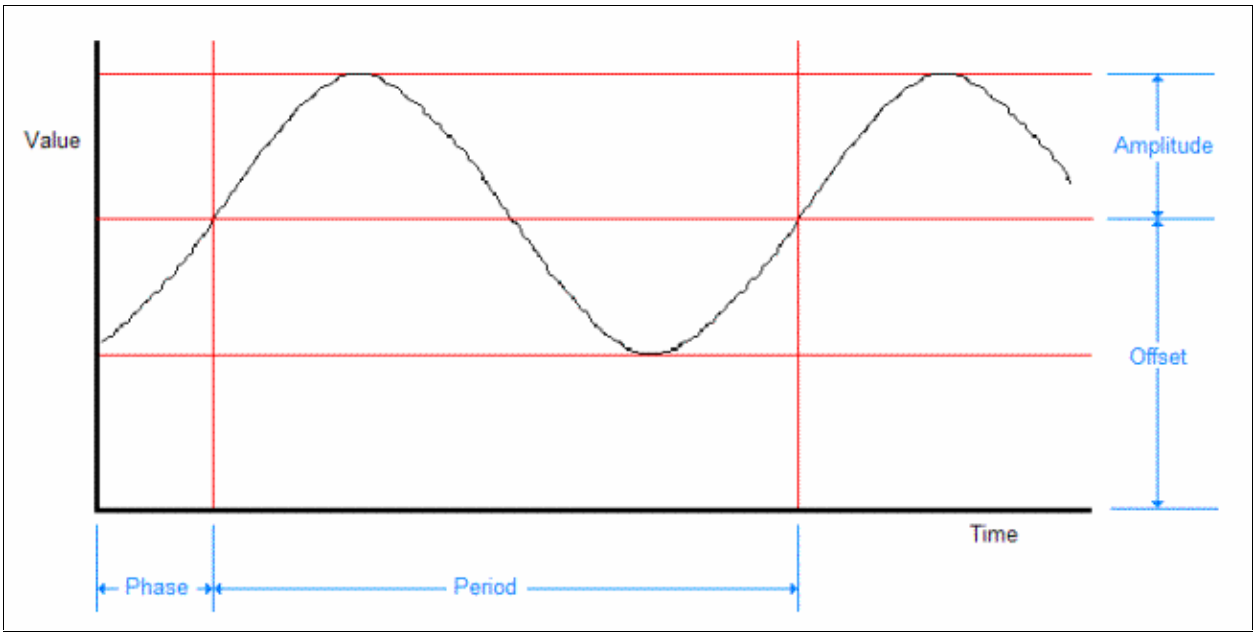

### *Ramp*

This function is a series of rising ramps. For a falling ramp, you can use the Triangle function with a Ratio set to zero.



## *Sine*

Mathematically, the sine function takes values between -1 and +1. This means that the value will vary both above and below the level set by Offset. Therefore, this is the only function for which the Offset parameter specifies the middle of the range, rather than the bottom. In addition, this is the only function for which the Amplitude parameter does not specify the peak-to-peak range of values.



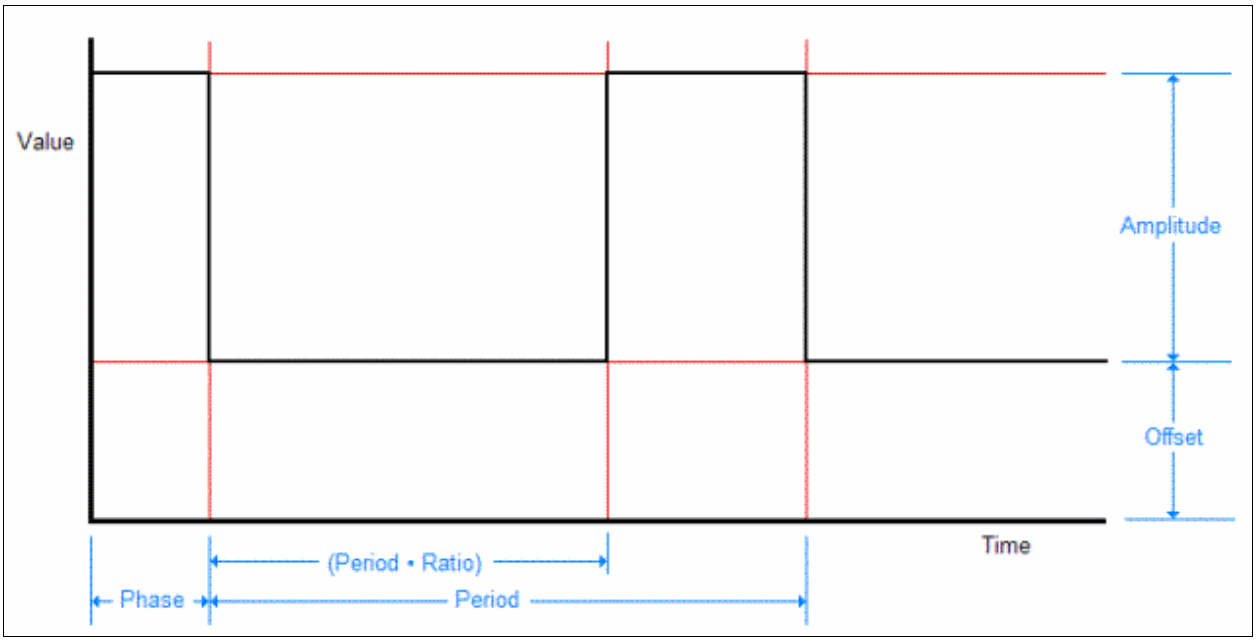
### *Square*

The Ratio parameter specifies the fraction of the cycle during which the value is low. This means that the relationship between Ratio and the duty cycle is:

Ratio = 1 – Duty Cycle

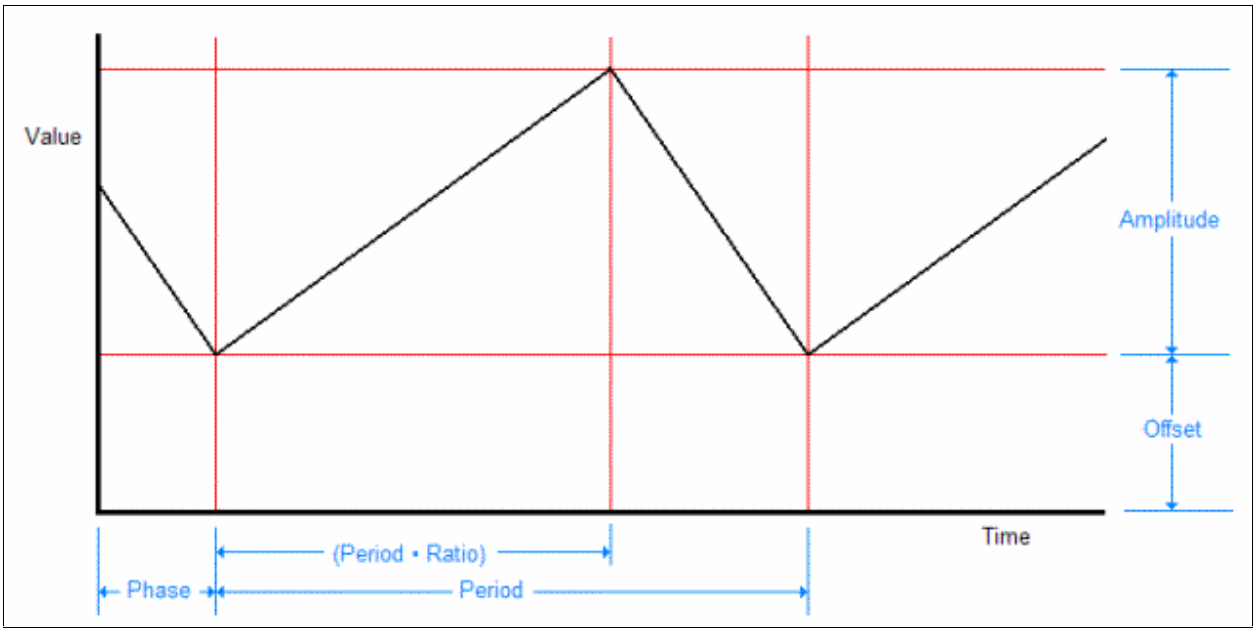
or

Duty Cycle = 1 – Ratio



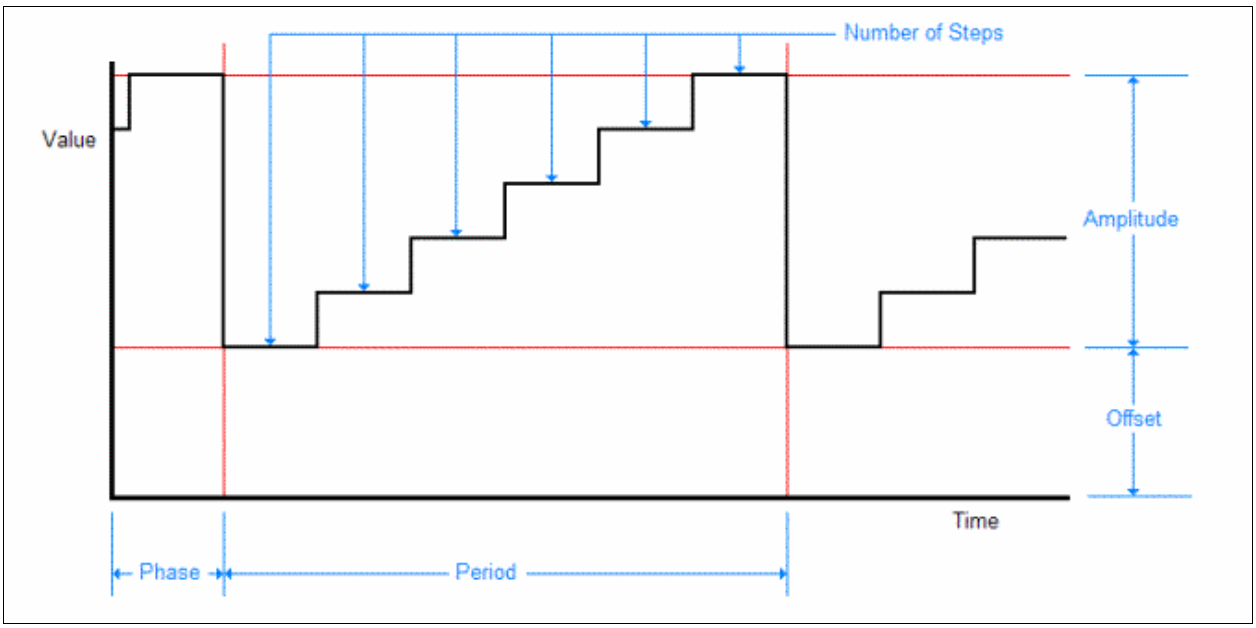
### *Triangle*

If the Ratio parameter is set to 1, this is the same as the Ramp function. If Ratio is set to 0, this will be a falling ramp function.




### *Step*

Notice that the Number of Steps parameter specifies the number of levels that the value will take, not the number of step increases. Because both the bottom and top steps are counted, there will be one fewer step increase than the Number of Steps. Keep this in mind when specifying this parameter.



# *Alarm Definitions*

The Cyberlogic OPC Server supports the OPC Alarms and Events specification. If your client application is also OPC AE compliant, it will then be able to receive the alarms and events reported by the Server. The user may define many different alarm conditions. A number of Data Items can then use each such condition. As a result, the user need not define the same alarm condition many times over.

There are two categories of alarms: digital and limit. Digital alarms are normally used for Boolean data and Limit alarms are normally used for numeric data, but either alarm type may be used with either data type. Alarms may not be used with string data, arrays or bit fields greater than 64 bits.

## Limit Alarms

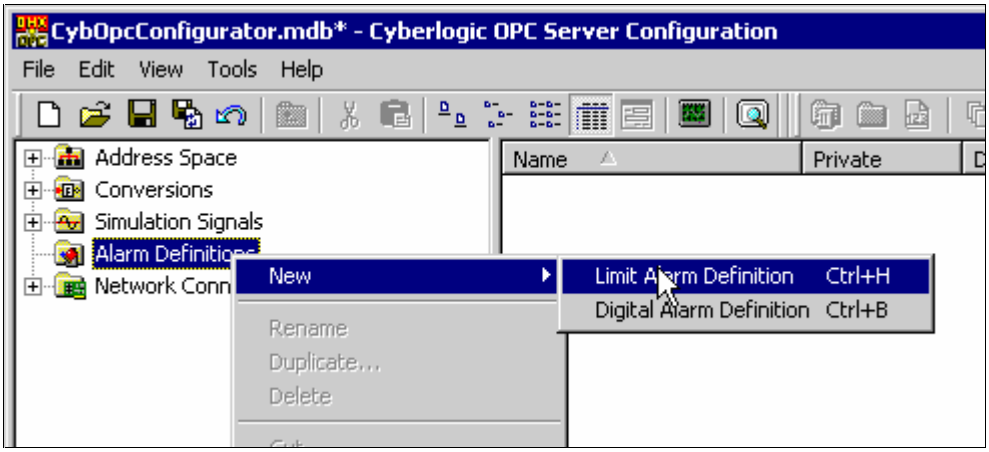
Limit alarms divide the range of values for the Data Item into five alarm states: LoLo, Lo, Normal, Hi and HiHi. These are normally used for numeric Data Items.

If you use a Limit alarm with a Boolean Data Item, the software will convert the value to numeric before checking for alarm conditions. A value of False will be converted to 0 and a value of True will be converted to -1.

Each alarm state allows you to designate a message and a severity level. You may also indicate whether the alarm requires a client-side acknowledgement. If you wish, you can specify a deadband value. The deadband prevents the server from generating a large number of alarm messages when the signal jitters around one of the limits.

## Creating a New Limit Alarm

Select the *Limit Alarms* root folder and select *New/Limit Alarm* from the Edit menu (or right-click on the *Limit Alarms* root folder and select *New/Limit Alarm* from the context menu). Enter the information required in the *Limit Alarm* dialog box, as described below. Click *Apply* when you are done.

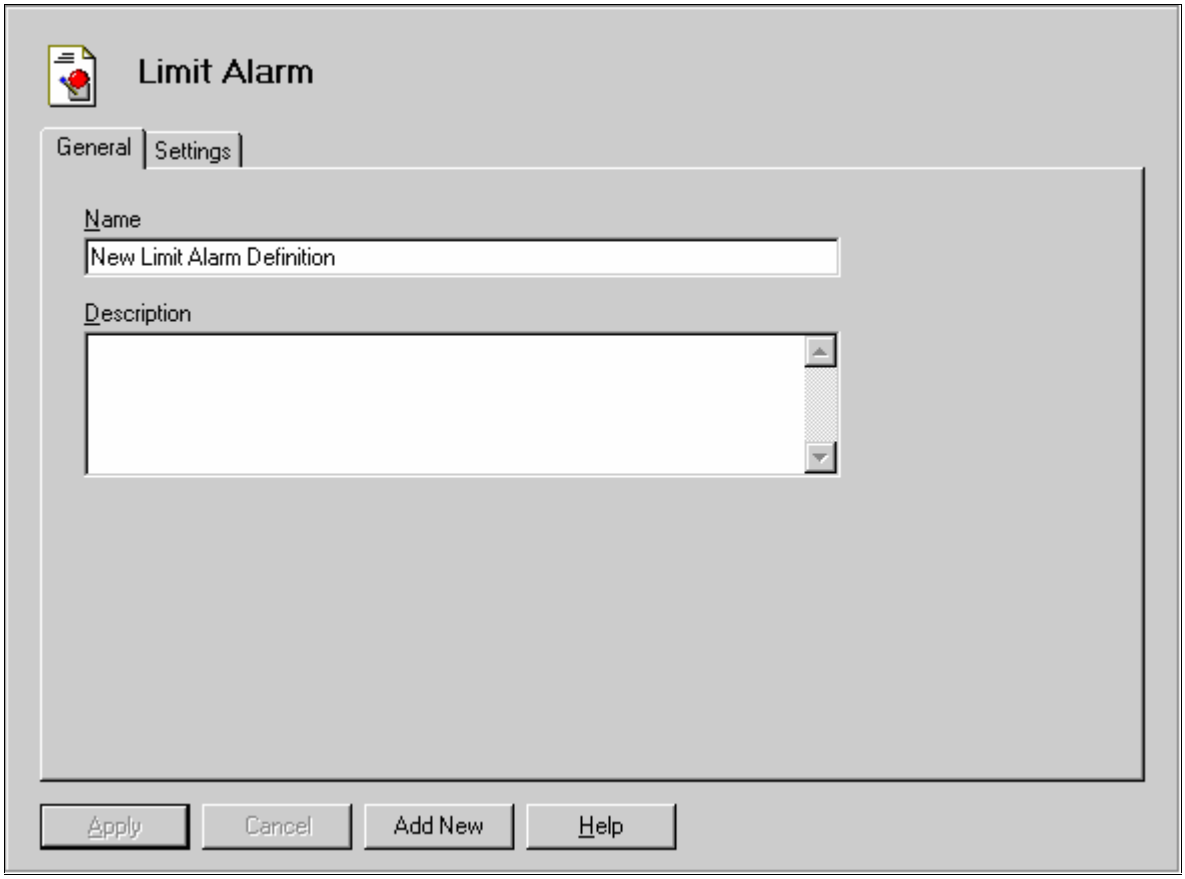


## Deleting a Limit Alarm

Select an existing limit alarm and press the *Delete* key (or right-click on a limit alarm and select *Delete* from the context menu).

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## General Tab



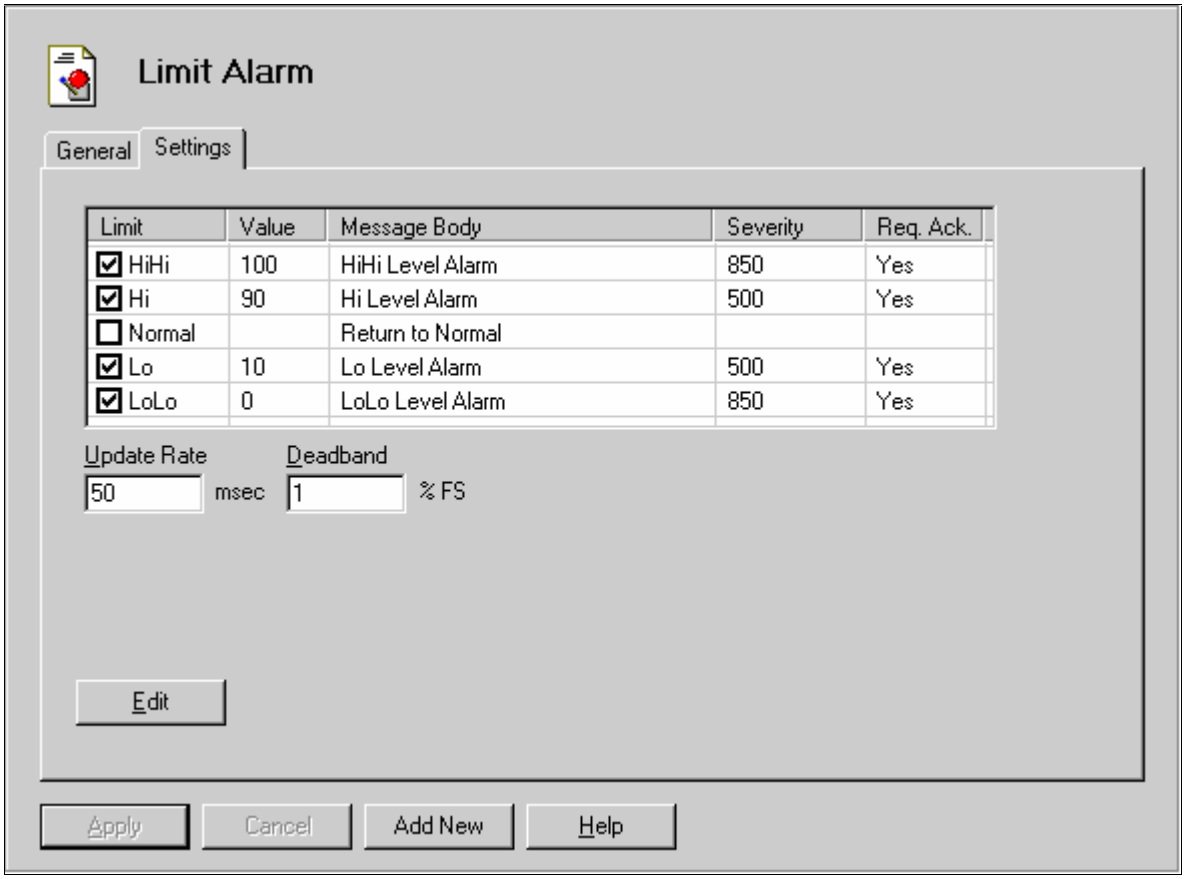

### *Name*

The Name identifies the Limit Alarm. It can be up to 50 characters long, may contain spaces, but must not begin with a space. It also must not contain any periods.

### *Description*

This optional field further describes the Limit Alarm. It can be up to 255 characters long.

## Settings Tab




### *Limit*

Check the boxes to indicate which alarm conditions the Server should report.

### *Value*

Enter the signal level that will trigger an alarm.

### *Message Body*

Enter the text that is associated with this alarm.

### *Severity*

The client uses this field to filter which events it wants to receive. Enter a value between 1 and 1000, where 1 is the least severe and 1000 is the most severe.

### *Req Ack.*

*Yes* indicates that an alarm must be acknowledged before it can clear.

### *Update Rate*

Enter an interval, in milliseconds, at which the Server will test the Data Item for an alarm state.

### *Deadband*

The Server will not reevaluate the alarm condition until the Data Item value changes by a minimum amount, specified by the Deadband value. Without a Deadband, a Data Item that jitters just above and then just below an alarm value will repeatedly trigger alarms, even though its value does not change significantly. The Deadband helps to prevent this from happening. Enter a value in percent of full scale.

## Digital Alarms

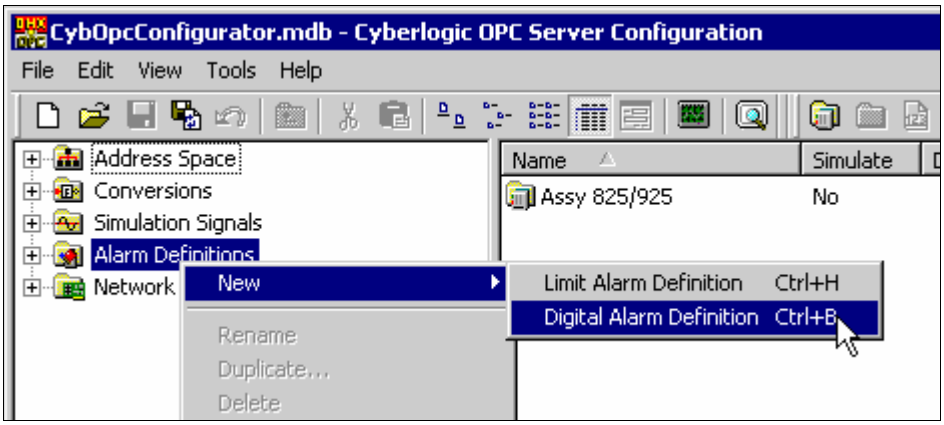
Digital Alarms specify an alarm to occur when the value is either True or False and so they are normally used with Boolean Data Items.

If you use a Digital Alarm with a numeric Data Item, a value of 0 is treated as False and any other value is treated as True.

Each alarm state allows you to designate a message and a severity level. You may also indicate whether the alarm requires a client-side acknowledgement. If desired, you can have the Server generate an alarm when the Data Item returns to its normal value.

## Creating a New Digital Alarm

Select the *Digital Alarms* root folder and select *New/Digital Alarm* from the Edit menu (or right-click on the *Digital Alarms* root folder and select *New/Digital Alarm* from the context menu). Enter the information required in the *Digital Alarm* dialog box, as described below. Click *Apply* when you are done.

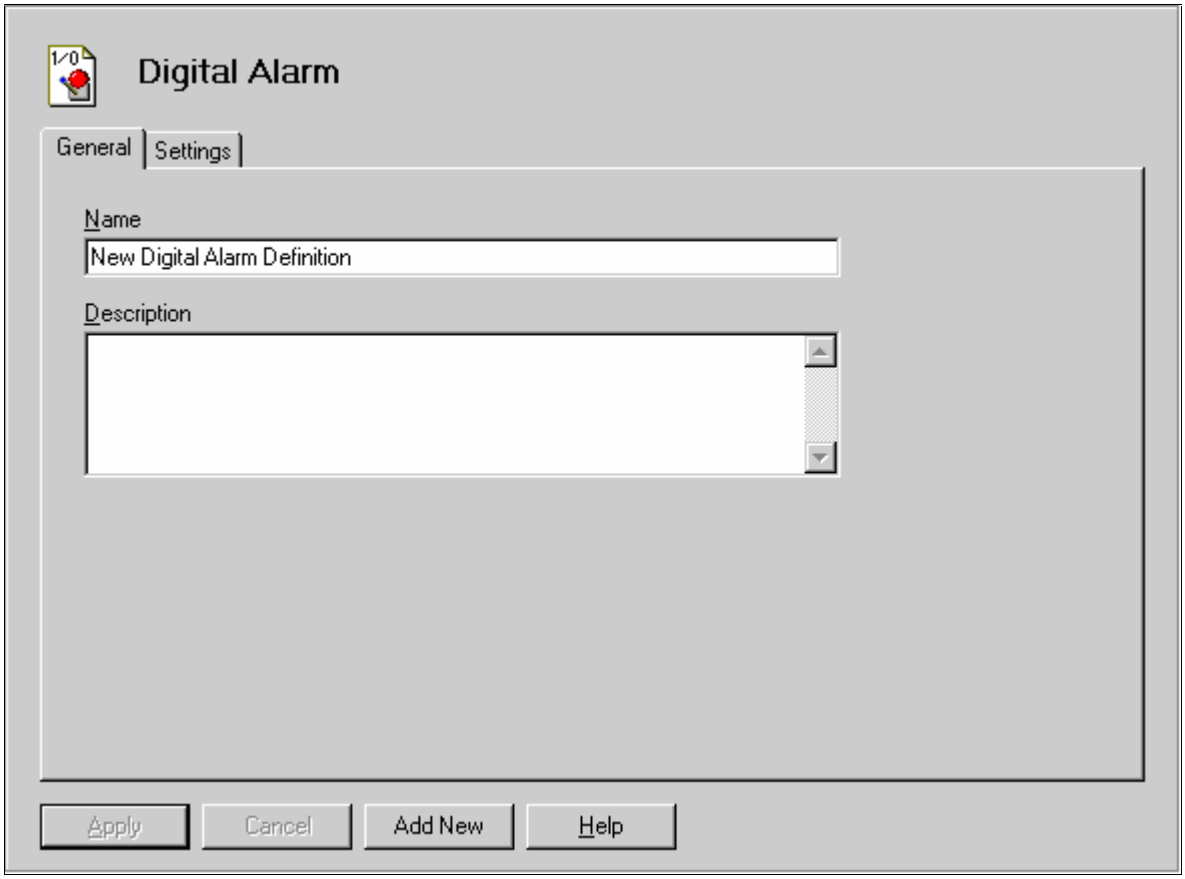



## Deleting a Digital Alarm

Select an existing digital alarm and press the *Delete* key (or right-click on a digital alarm and select *Delete* from the context menu).

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## General Tab



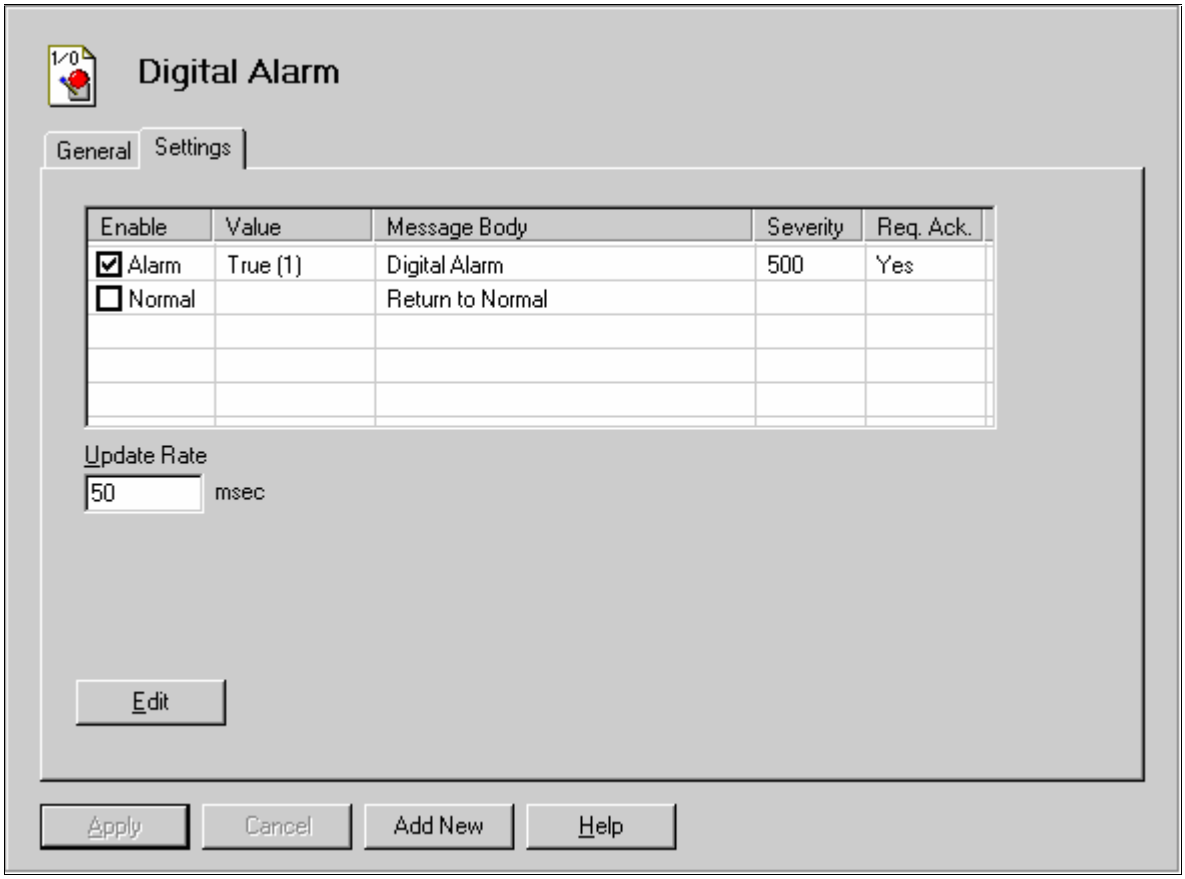

### *Name*

The Name identifies the Digital Alarm. It can be up to 50 characters long, may contain spaces, but must not begin with a space. It also must not contain any periods.

### *Description*

This optional field further describes the Digital Alarm. It can be up to 255 characters long.

## Settings Tab




### *Enable*

Check the box to indicate which alarms the Server should report.

### *Value*

Select either True (1) or False (0) as the Data Item state that will trigger an alarm.

### *Message Body*

Enter the text that is associated with this alarm.

### *Severity*

The client uses this field to filter which events it wants to receive. Enter a value between 1 and 1000, where 1 is the least severe and 1000 is the most severe.

### *Req Ack.*

*Yes* indicates that an alarm must be acknowledged before it can clear.

### *Update Rate*

Enter an interval, in milliseconds, at which the Server will test the Data Item for an alarm state.

# *Saving Configuration Changes*

The Cyberlogic OPC Server Configuration Editor keeps track of recent configuration changes. Until you save these changes, you can revert to the previously saved configuration. The Editor supports two types of save operations. The standard Save operation saves the changes without updating the Server or the connected clients. The Save & Update Server operation saves the changes and also updates the Server and all connected clients.

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

### Saving Configuration Without Updating the Server

To save the configuration without updating the Server, select *Save* from the File menu (or click the *Save* button on the standard buttons toolbar). The changes will be saved but the Server will still be running with the old configuration.

### Saving Configuration and Updating Server

To save the configuration and update the server, select *Save & Update Server* from the File menu (or click the *Save & Update Server* button on the standard buttons toolbar).

# *Undoing Configuration Changes*

To undo configuration changes and revert to the previously saved configuration, select *Undo Changes* from the File menu (or click the *Undo Changes* button on the standard buttons toolbar).

# *Configuration Import/Export*

The Cyberlogic OPC Server normally stores its configuration information in a binary database file. The Cyberlogic OPC Server Configuration Editor and the run-time Server module can easily read and operate on this file. However, a different file format may be preferred for quick viewing or processing by other applications.

The Export feature allows the entire Server configuration to be saved in three text formats: comma delimited (.csv), tab delimited (.tab) and XML.

The Import feature allows you to import these exported files or selected portions of them. You may also import configurations from other vendors' OPC servers.

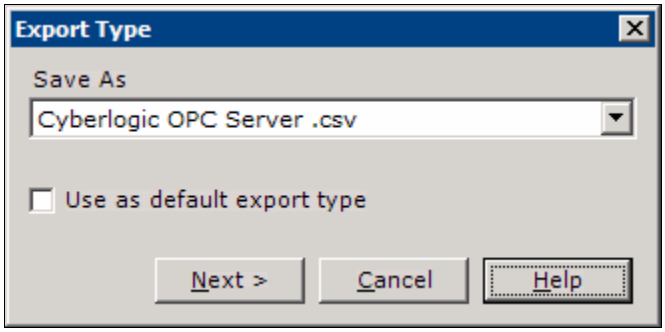
The Import/Export capability of the Cyberlogic OPC Server Configuration Editor reduces the time and effort required to configure the Server. In addition, it can speed up the configuration of similar Cyberlogic OPC Servers and is useful for backing up and restoring Server configurations. Furthermore, because it

can import configurations from other OPC server brands, it can greatly reduce the effort needed to migrate from those servers to Cyberlogic's Server.

## Exporting a Server Configuration

This feature exports an entire Server configuration to one of the supported file formats. To do this, use the following procedure:

1. Select *Export…* from the *File* menu. You will see the following dialog.

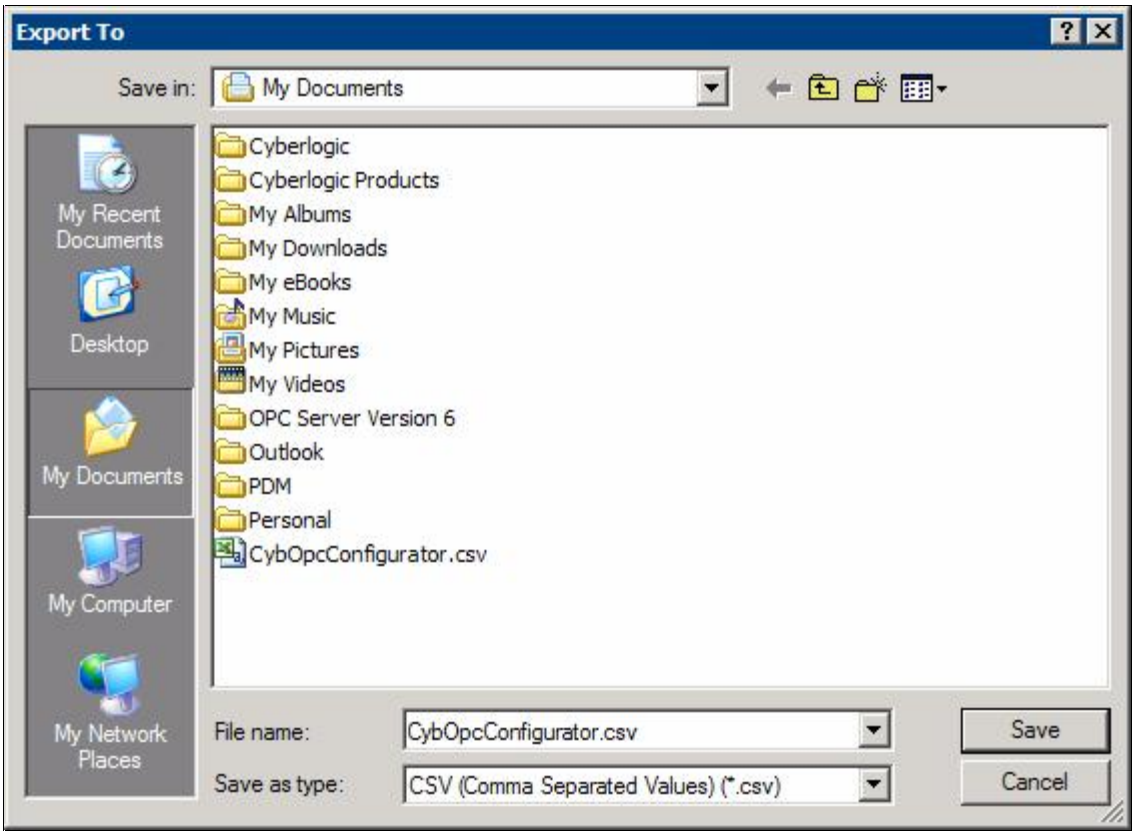



From the drop box, you can select the format to use for the exported file. Your choices are:

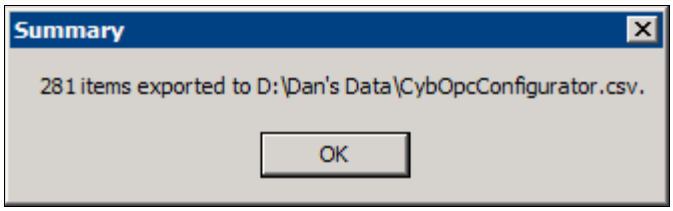
- Comma separated values
- Tab separated values
- XML file

If you check *Use as default export type*, your selected type will be the default selection when you run the export utility in the future.

2. Click *Next* > and the following screen will appear. Select the desired directory and enter a file name.



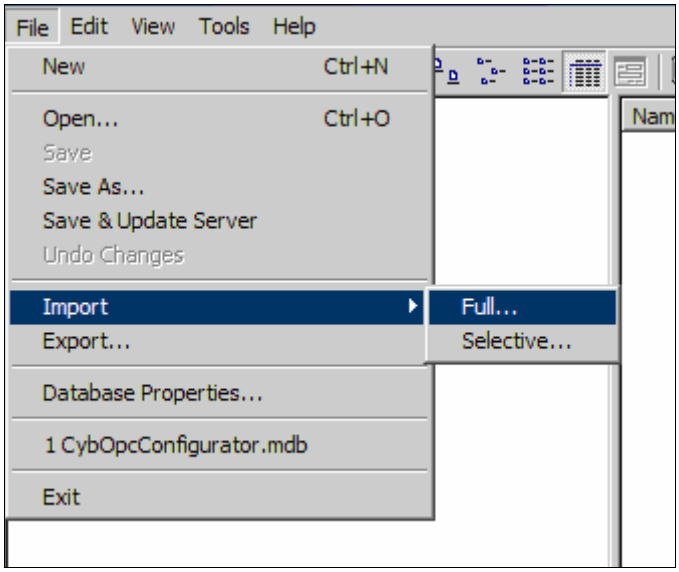

3. Click *Save* to create the export file. The following message will notify you about the number of exported items. Click *OK* to complete the process.



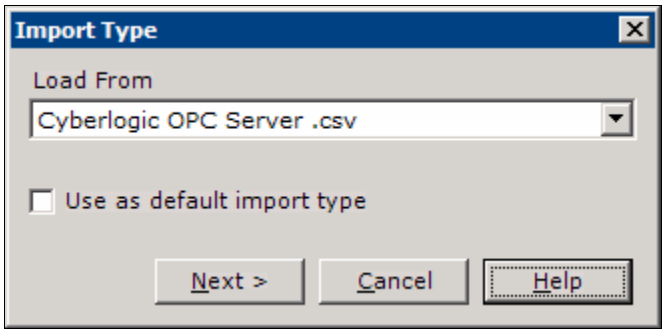
## Importing an Entire Server Configuration

You can import all or part of a Server configuration from a previously exported file or from certain other configuration file formats. To import an entire file, use the following procedure:

1. Select *Import/Full…* from the File menu.




2. You will see the following dialog.

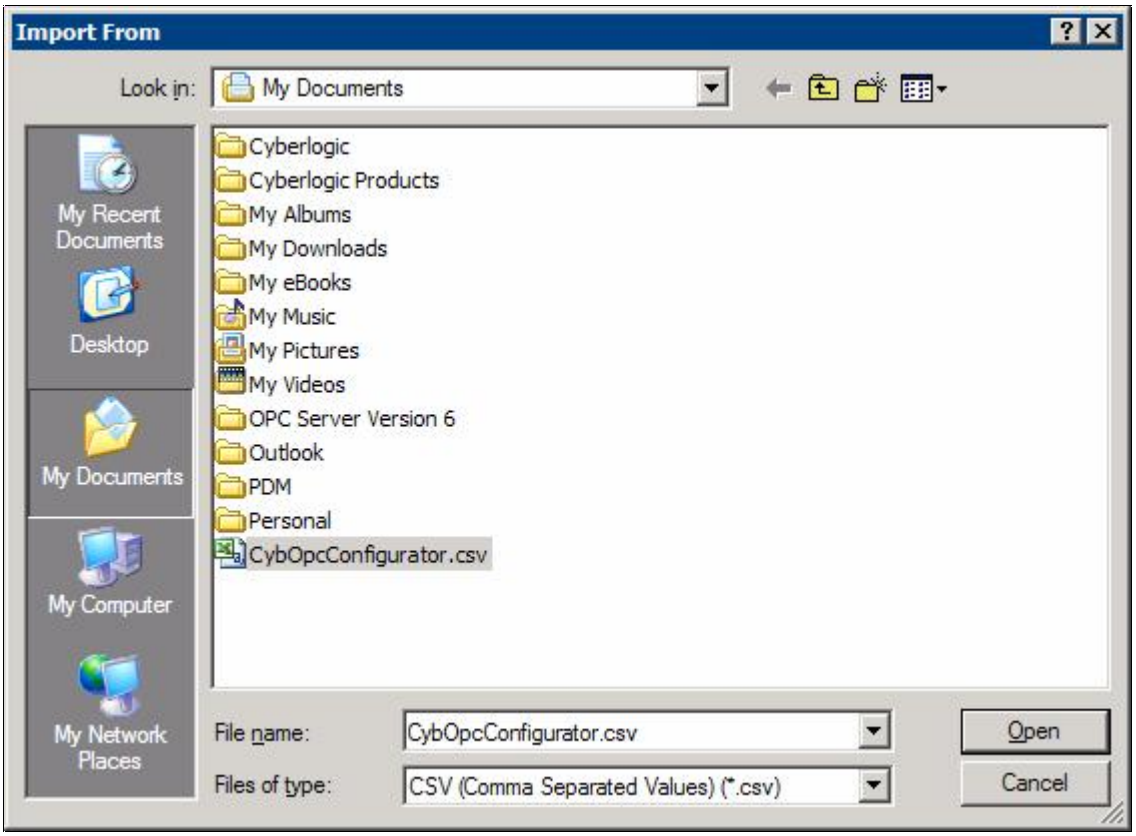



From the drop box, you can select the format of the file you wish to import. Your choices are:

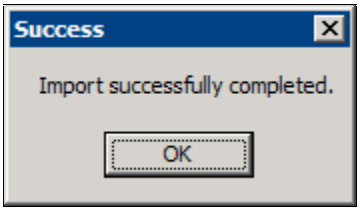
- Cyberlogic OPC Server .mdb (Access database)
- Cyberlogic OPC Server .csv (Comma separated values)
- Cyberlogic OPC Server .tab (Tab separated values)
- Cyberlogic OPC Server .xml (XML file)
- Various other vendors' OPC server formats, depending on the driver agents you have installed

If you check *Use as default import type*, your selected type will be the default selection when you run the import utility in the future.

3. Click *Next* > and the following screen will open. Choose the file containing the configuration you wish to import and click *Open*.




4. Progress dialogs will open to show the status of the import. When it is finished, the Success dialog will be displayed.




5. Click *OK* to complete the operation.

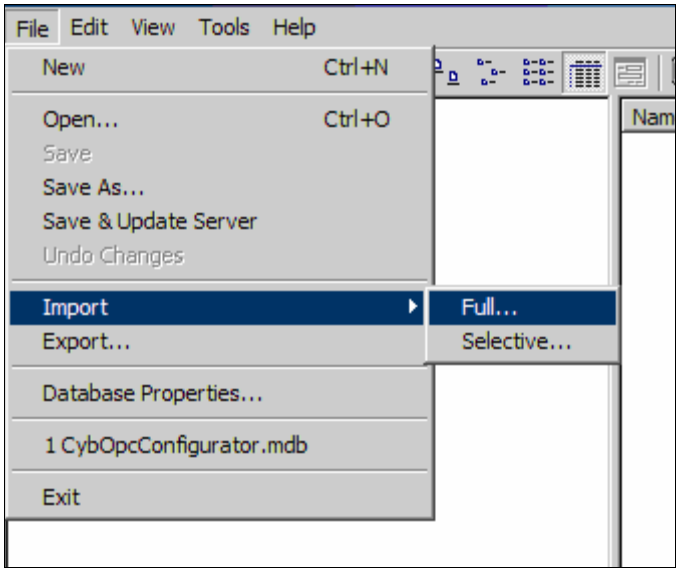
| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

## Importing Configurations From Other Server Brands

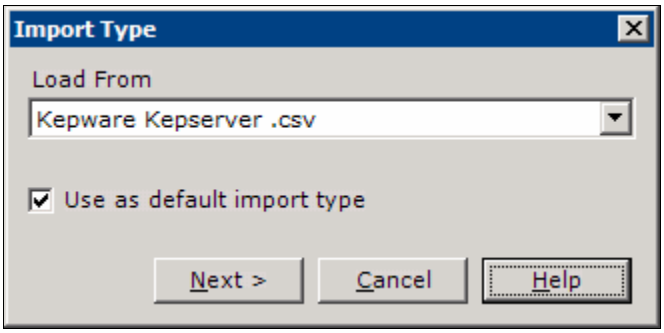
Importing a configuration from a non-Cyberlogic OPC server is handled in the same manner as the import described above. However, because of variations in features between the different brands of servers, there may be additional conflicts and errors to deal with.

This example shows how to import an entire Kepware server configuration. Before you begin the import process, you must use the Kepware tools to export the configuration to a csv file.

1. Select *Import/Full…* from the File menu.

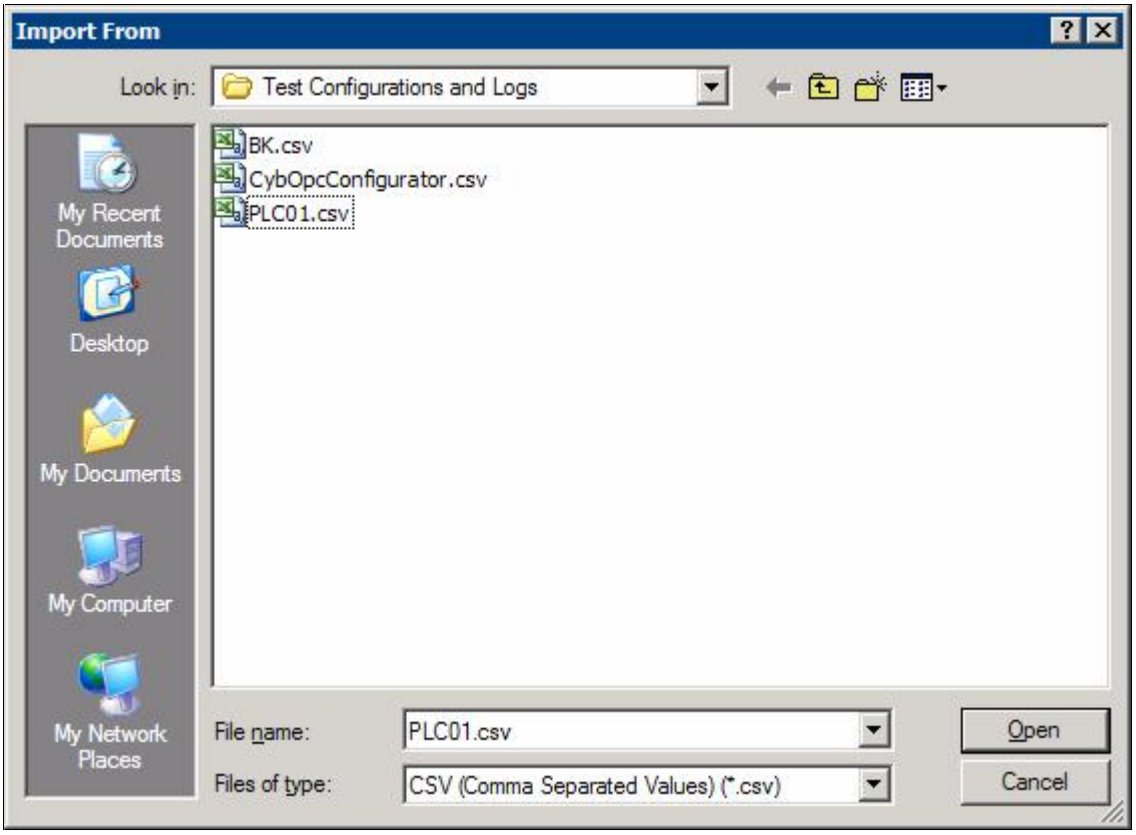


2. You will see the following dialog.




Select the format of the file you wish to import, in this case you would choose *Kepware Kepserver .csv*.

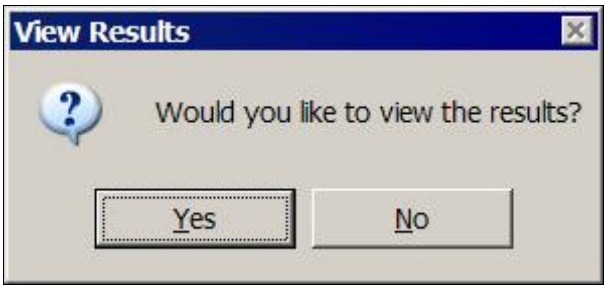
3. Click *Next* > and the following screen will open. Choose the file containing the configuration you wish to import and click *Open*.



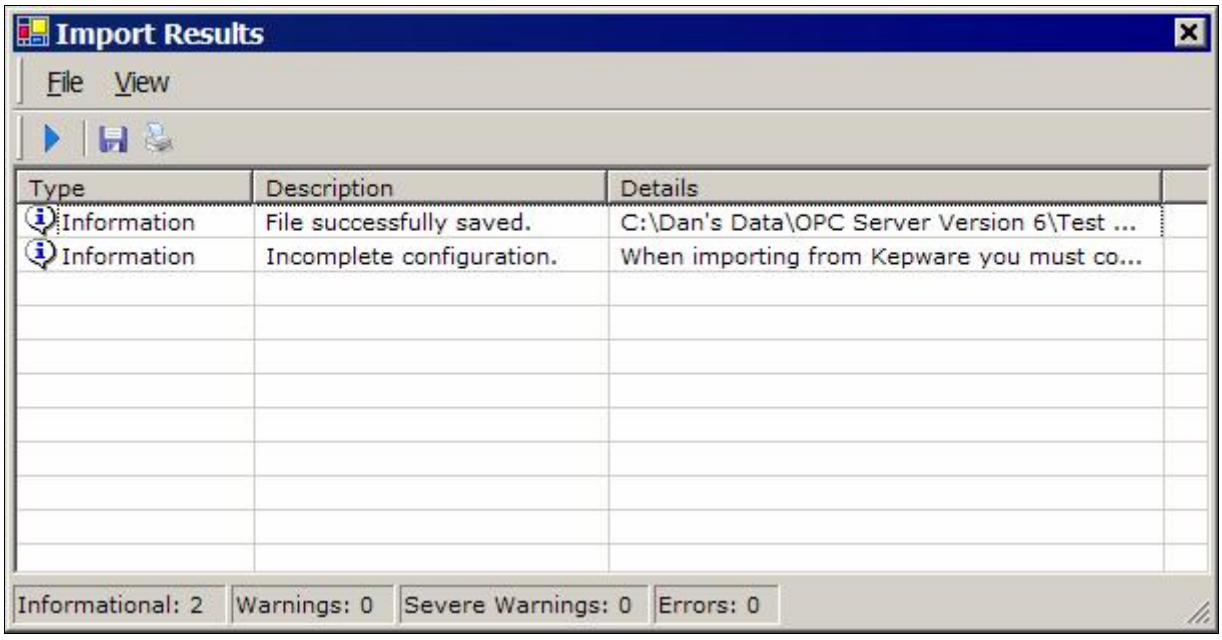
4. The next screen will ask you to specify the brand of the controller whose configuration you are importing. Select the proper brand and click *OK*.


5. The import utility will partially process the file and then will ask you to choose the specific controller family. Make your selection and then click *OK*.

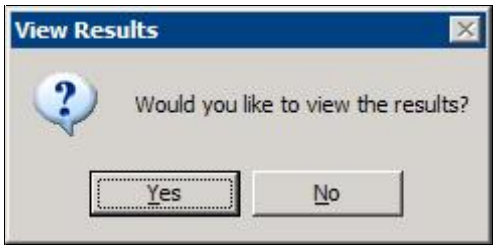

6. The utility will finish preprocessing the file. It will then give you the chance to view the results, which may contain warnings of errors or other useful information.



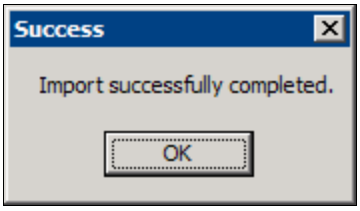

7. The results screen will show you the errors, warnings and other information with a description of each. Click  to continue with the import processing.



8. After the utility finishes the import processing, it will ask if you wish to view the results. Click *Yes*.




9. You will be shown the results screen with all of the detected errors and warnings. After you exit from that screen, the utility will acknowledge completion of the operation. Click *OK*.

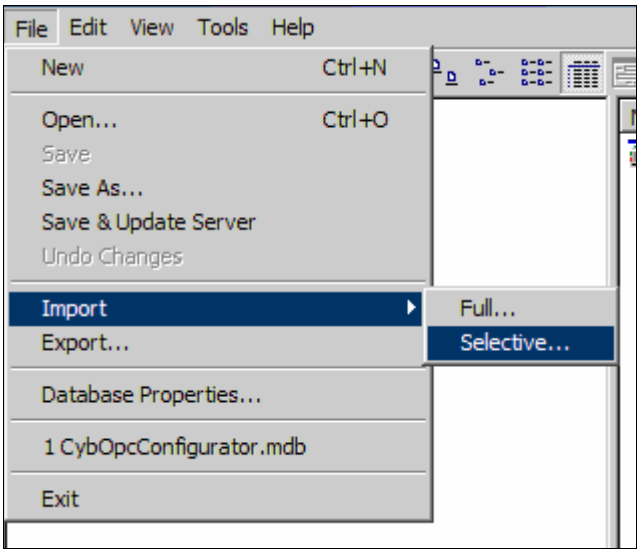



| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

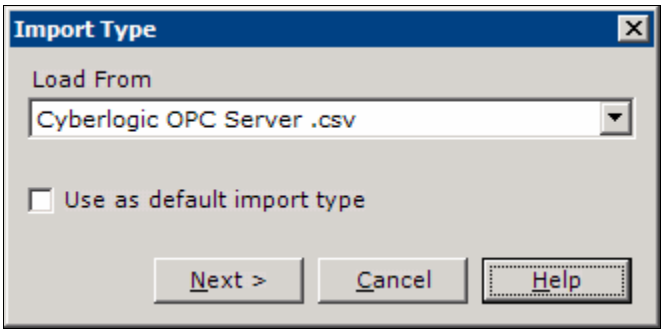
## Importing a Partial Server Configuration

To import a part of a source file, use the following procedure. Partial imports are available when importing from Cyberlogic or other configurations. For this example, we will use a Cyberlogic configuration.

1. Select *Import/Selective…* from the *File* menu.



2. You will see the following dialog:

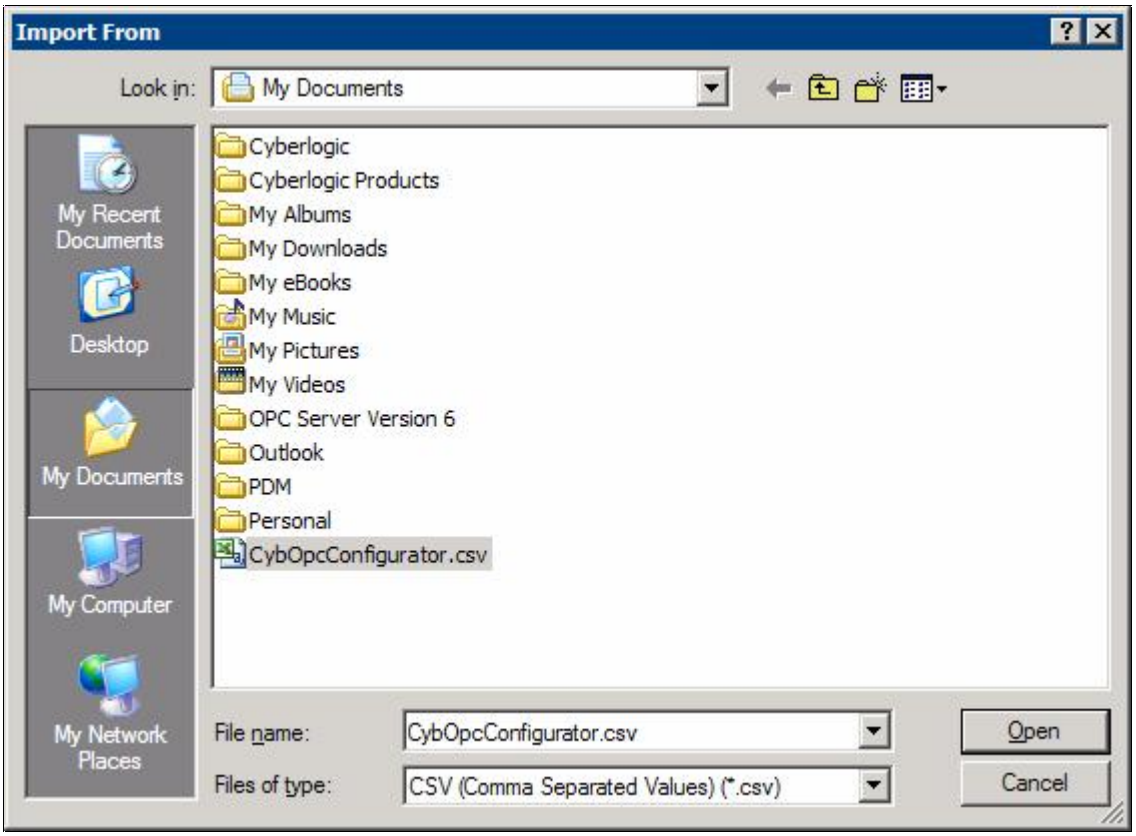



From the drop box, you can select the format of the file you wish to import. Your choices are:

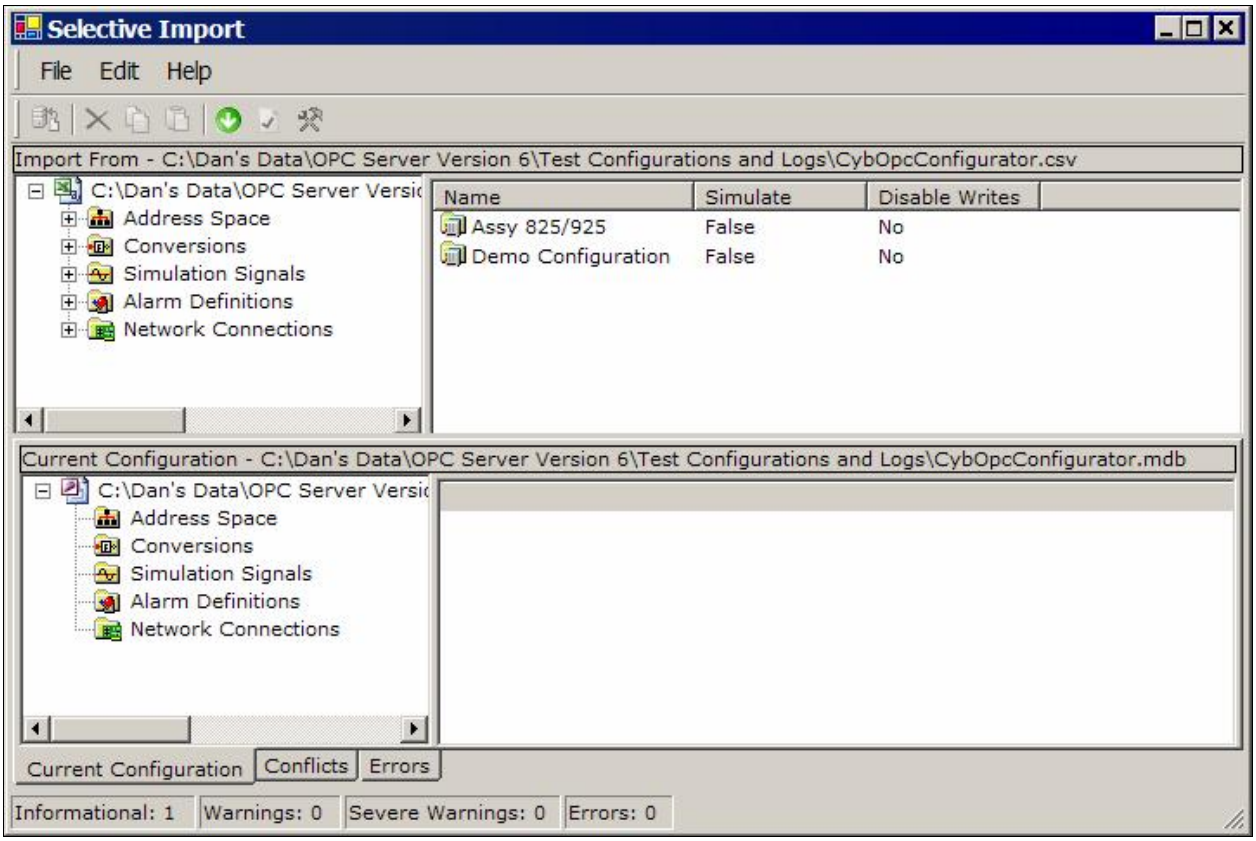
- Cyberlogic OPC Server .mdb (Access database)
- Cyberlogic OPC Server .csv (Comma separated values)
- Cyberlogic OPC Server .tab (Tab separated values)
- Cyberlogic OPC Server .xml (XML file)
- Various other vendors' OPC server formats, depending on the driver agents you have installed

If you check *Use as default import type*, your selected type will be the default selection when you run the import utility in the future.

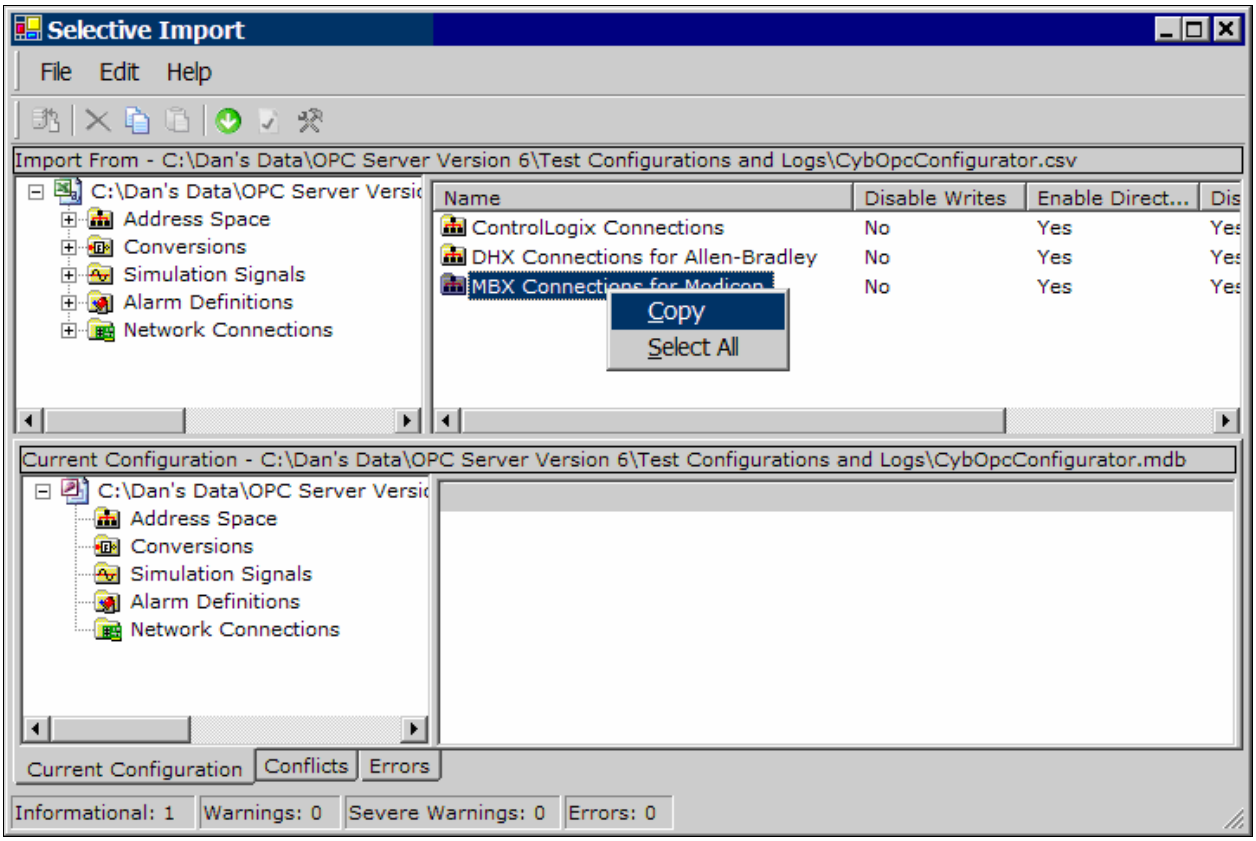
3. Click *Next* > and the following screen will open. Choose the file containing the configuration you wish to import and click *Open*.



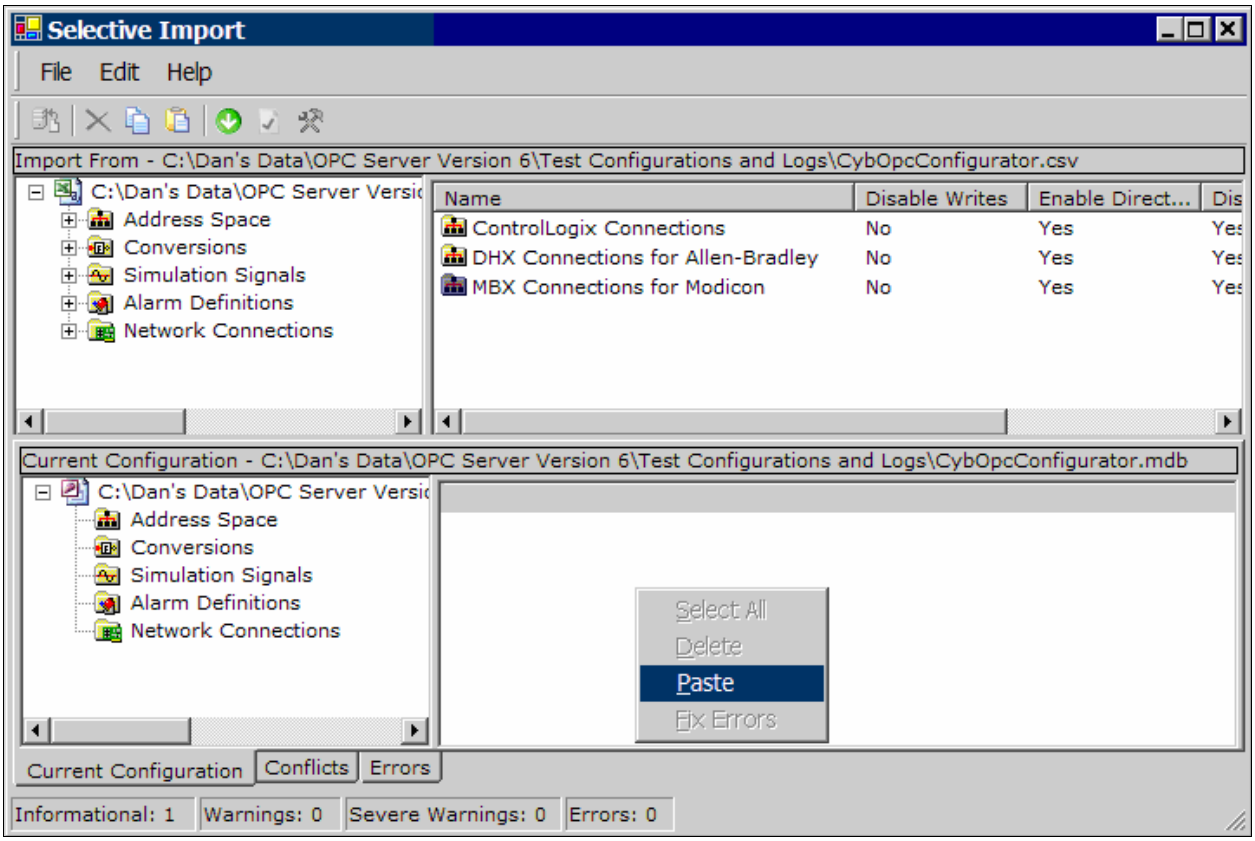
4. The Selective Import editing screen will open.



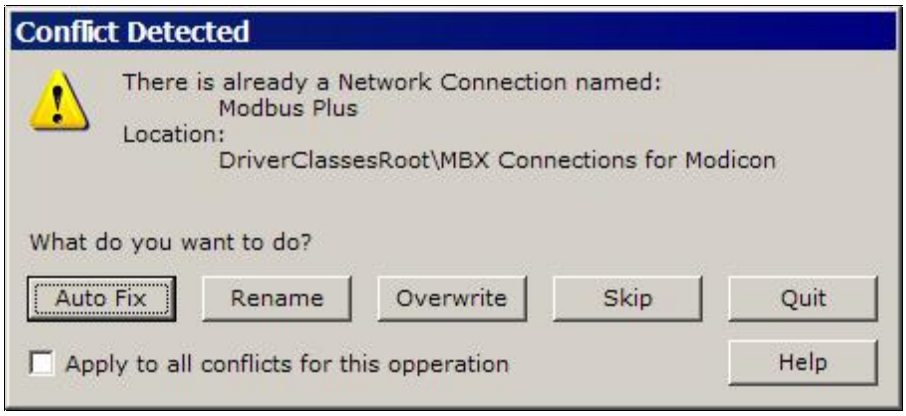
5. The upper pane shows the configuration you are importing from and the lower pane shows the current configuration of your server. Navigate through the *Import From* tree to find the items you wish to import. Highlight them, then right-click and select *Copy*.



6. Navigate to the proper place in the lower pane, then right-click in the pane and select *Paste*. The copied item will be imported into the server configuration.



7. If one or more items you are trying to paste already exist in your server's current configuration, the Conflict Detected dialog will appear.




You must choose how to handle the conflicts. Your choices are:

- *Auto Fix:* The editor will rename the item so that it does not conflict with any other names.
- *Rename:* You must enter a new name that will not conflict.
- *Overwrite:* The item you are pasting will replace the conflicting item in your current configuration.
- *Skip:* The editor will keep the item in your current configuration and not import the item you pasted.
- *Quit:* The paste operation will terminate immediately. Any items that were added on this paste operation before the editor detected the conflict will remain, but the conflicting item and all remaining items that were to be pasted will not be imported.

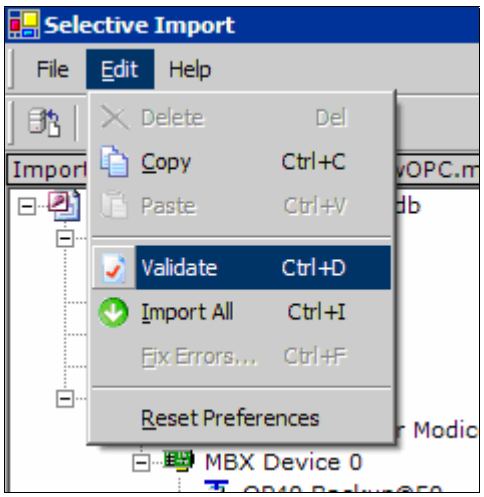
If you selected more than one item to paste and you check *Apply to all conflicts* before selecting *Auto Fix, Rename, Overwrite* or *Skip*, then that selection will apply to all of the conflicting items detected for that paste operation.

8. You must now validate and save the configuration, as described in the next section.

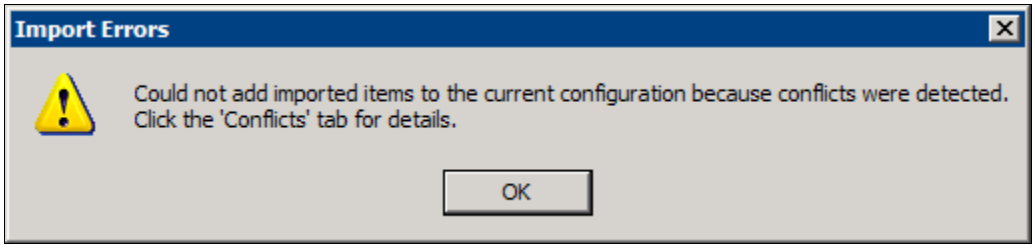
## Validating and Saving an Imported Configuration

Once you have imported all of the items you wish to add to your configuration, you must first validate them before you can save the configuration.

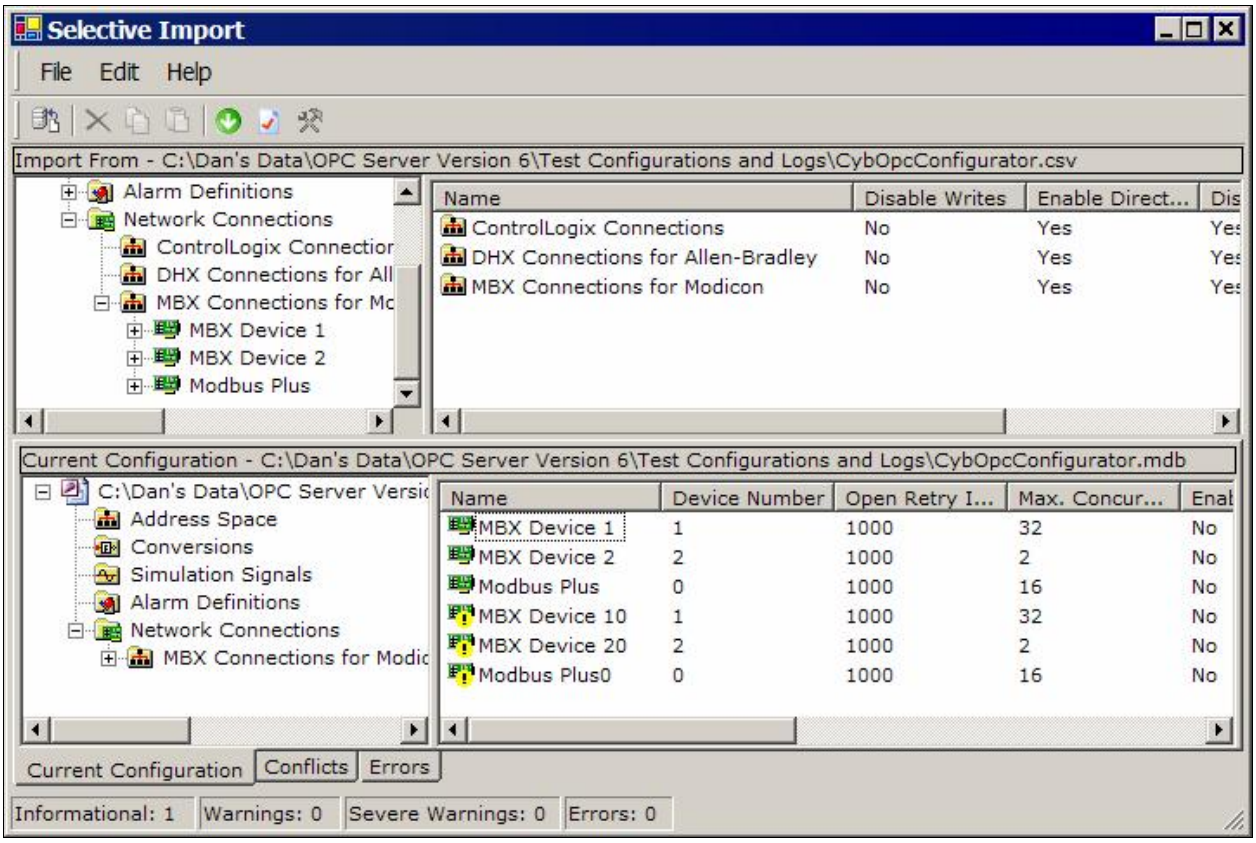
1. From the *Edit* menu, select *Validate*.



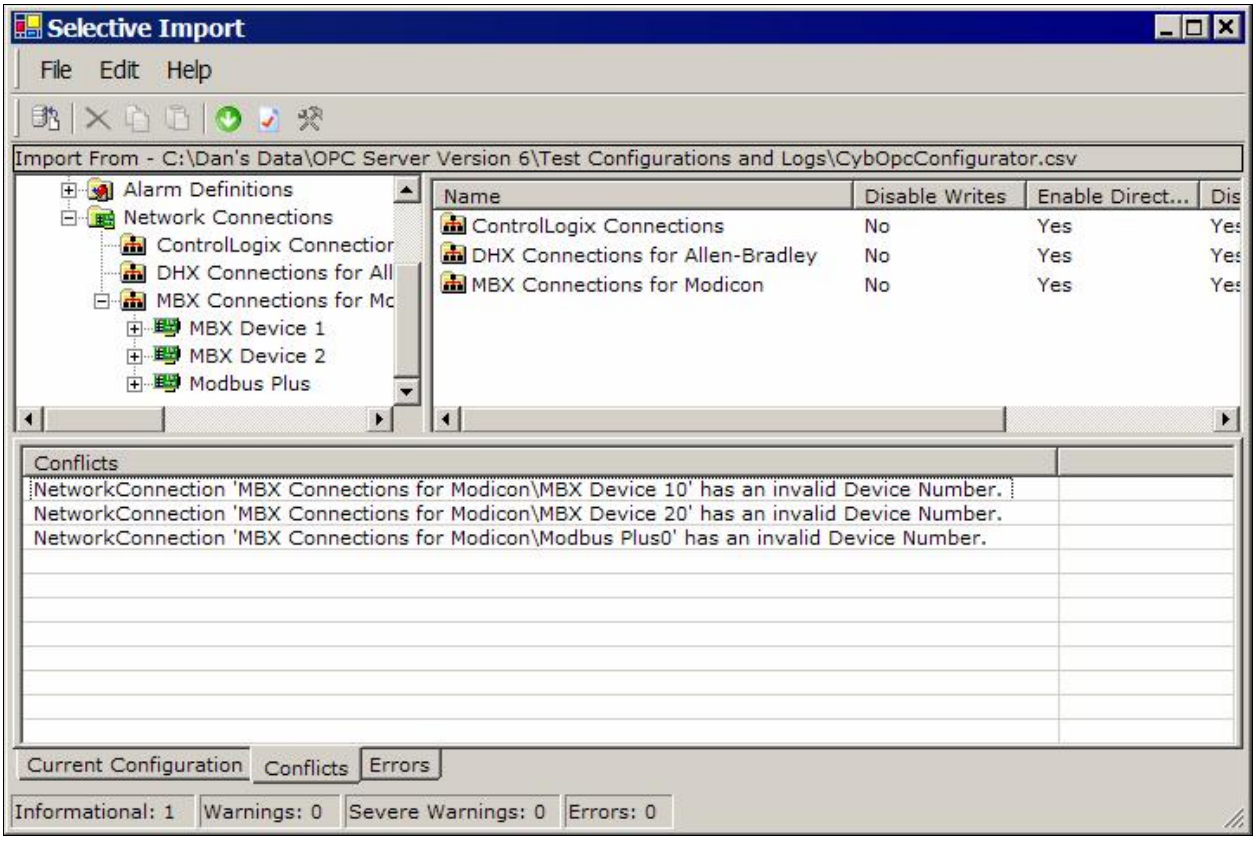

2. The editor will perform the validation and will inform you of any conflicts.



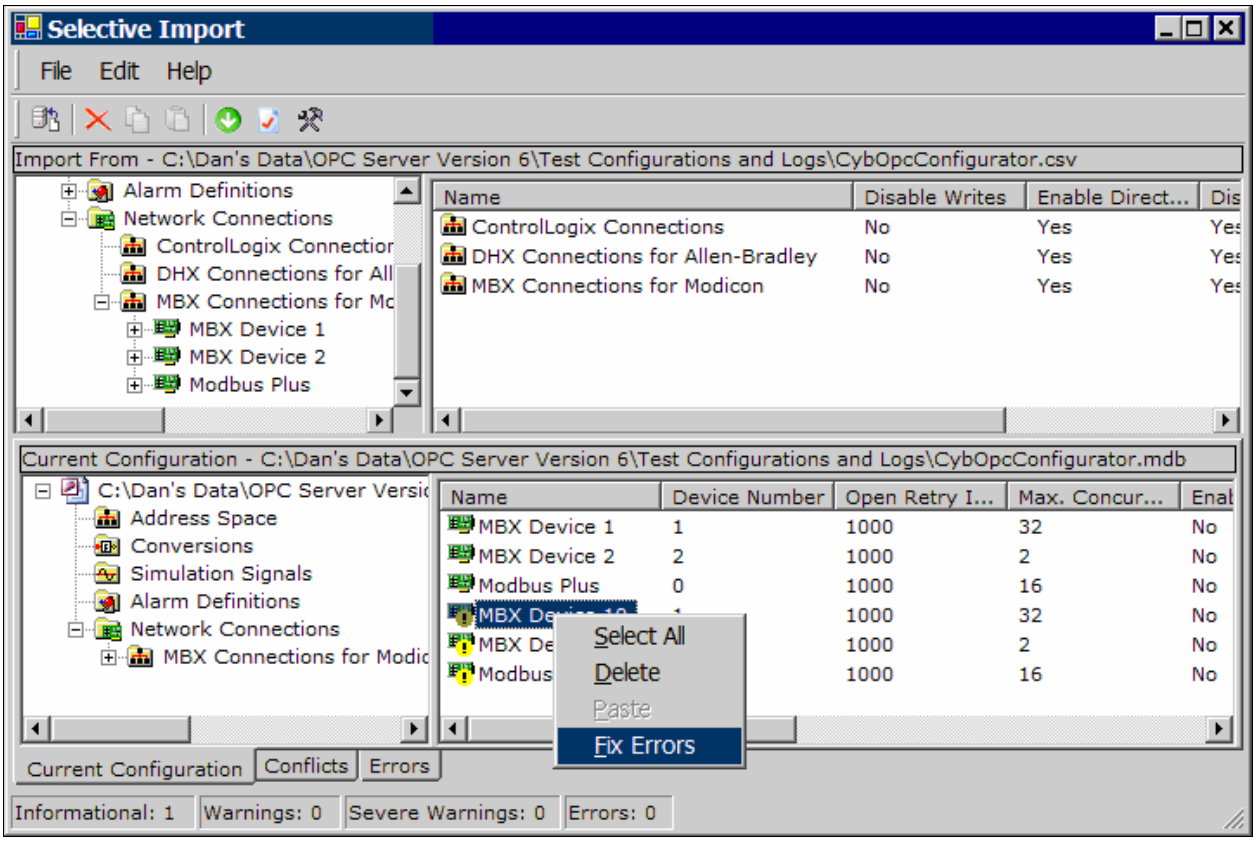
3. The items that failed the validation will be highlighted with the yellow exclamation point symbol. Click the *Conflicts* tab for an explanation of the problem.



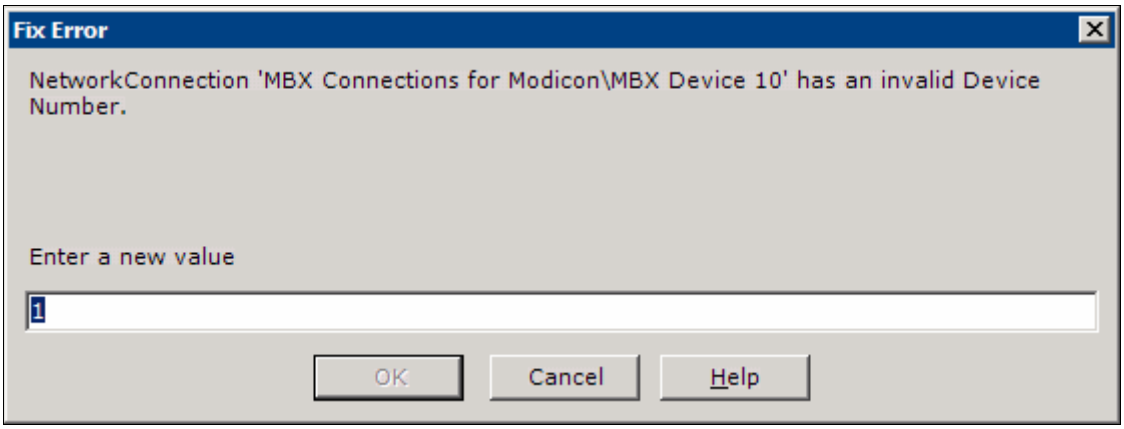
4. In this instance, there are duplicate device numbers. Notice from the previous figure that the imported item *MBX Device 10* has the same device number as *MBX Device 1*. You must change one of these device numbers to resolve the conflict. You must also resolve the other two conflicts that the validator detected.



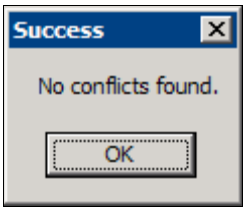
5. Right-click on the problem item and select *Fix Errors*.



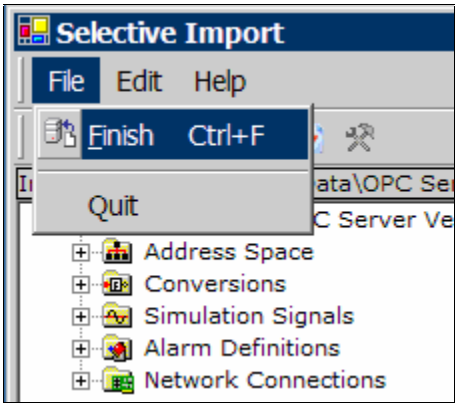

6. The Fix Error dialog will help you to correct the problem. In this case, enter the new address and click *OK*.



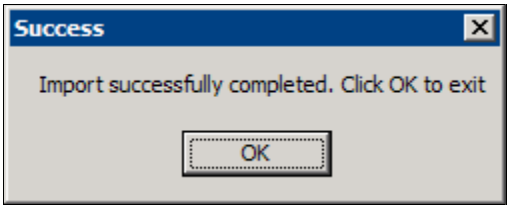

7. After you have cleared all of the errors, repeat the validation operation. When it succeeds, you will get a confirmation dialog. Click *OK*.



8. From the File menu, select *Finish*.




9. The import operation completes, your configuration is saved and the confirmation dialog appears. Click *OK* to exit from the Import utility.



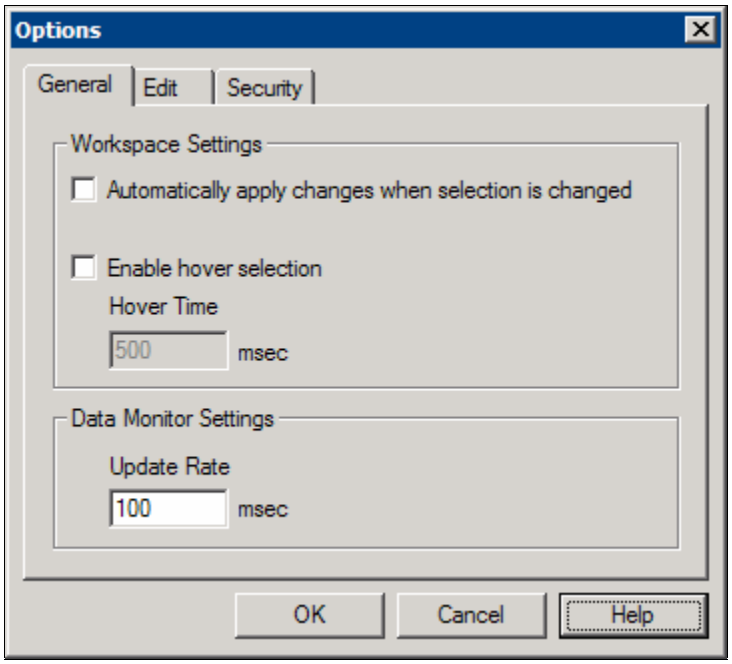

10. From the File menu, select *Save & Update Server*.

| | |
|---|---|
| **Caution:** | After you edit the configuration, you must open the *File* menu and select *Save & Update Server* for the changes you have made to take effect. Otherwise, the Server will still be running with the old configuration. |

# *Options*

You can access several Editor options by going to the Tools menu and selecting *Options*….

## General Tab




### *Automatically Apply Changes When Selection Is Changed*

When this box is not checked and you make changes to the configuration then select another object, the Editor will ask you to confirm that you want to save the changes. If you check the box, the Editor will save the changes automatically, without asking.

### *Enable Hover Selection*

When this box is not checked, you must click on the items in the right pane of the Selective Import screen to select them. If you check the box, you can select an item just by pointing to it with the mouse.

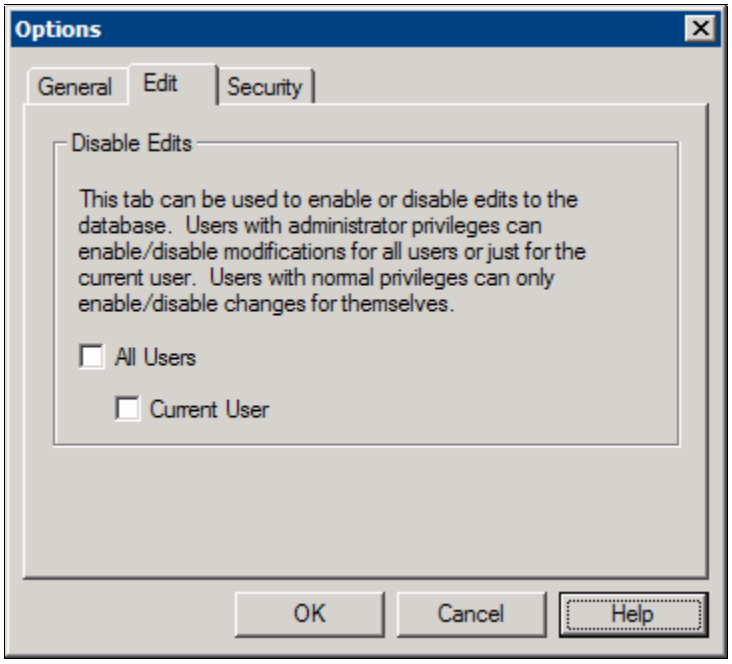
### *Hover Time*

When Hover Selection is enabled, this is the amount of time, in milliseconds, that the mouse pointer must hover before the item is selected automatically.

### *Update Rate*

Here you can specify the rate, in milliseconds, at which the Data Monitor will request data updates.

## Edit Tab




This tab allows you to enable and disable configuration edits. Users with administrator privileges can enable and disable edits for all users or just for the current user. Users with normal privileges can enable and disable changes only for themselves.

## Security Tab

For an OPC client and server to communicate, both systems must have proper DCOM security settings. There is no standard setting for OPC applications because each installation's security needs are unique. Therefore, you must decide how to configure security for your systems. For help in making these decisions and a detailed discussion of how to do the configuration, refer to the document *OPC & DCOM: A Guide to Using the Cyberlogic OPC Server via DCOM*. A copy of this document was installed on your system along with the software, and can be accessed from the Windows Start menu by navigating to the OPC Server directory and clicking on *DCOM Help*.

As part of the security configuration, you must select the access permissions, authentication level and impersonation level to be used. Two settings for these parameters are used as defaults by many OPC-based products. These may be selected by choosing the *Low* or *Medium* settings from this screen. If neither of these is the correct setting for your situation, you must choose *Custom* and click *Launch DCOMCNFG...* to configure all of the security settings yourself through the Windows operating system.

| | |
|---|---|
| **Caution:** | The preconfigured *Low* and *Medium* security settings override only the access permissions, authentication level and impersonation level for the Cyberlogic OPC Server. The rest of the security settings must still be configured with the DCOMCNFG utility. |


| | |
|---|---|
| **Caution:** | If you change the selection on this tab, you must restart the Cyberlogic OPC Server for the new settings to take effect.<br>To restart the Server, open the Windows Control Panel, go to Administrative Tools and then Services. Right-click on *Cyberlogic OPC Server* and select *Restart*. |

If the *Low* or *Medium* security settings match your client application's settings, select the appropriate radio option and click the *OK* button.

### *Low*

The *Low* level of security overrides the default server-specific security values, giving them the following settings:

| Security Parameter | Setting | Description |
|---|---|---|
| Access Permissions | All Users | Allows calls from anyone. |
| Authentication Level | None | No authentication occurs. |
| Impersonation Level | Identify | The server can obtain the client's identity. The server can impersonate the client to do access control list (ACL) checks, but it cannot access system objects as the client. |

**Note:** When the Server starts, it will set the security level with the following call:

```
CoInitializeSecurity(
NULL,
-1,
NULL,
NULL,
RPC_C_AUTHN_LEVEL_NONE,
RPC_C_IMP_LEVEL_IDENTIFY,
NULL,
EOAC_NONE,
NULL);
```

### *Medium*

The *Medium* level of security overrides the default server-specific security values, giving them the following settings:

| Security Parameter | Setting | Description |
|---|---|---|
| Access Permissions | All users | Allows calls from anyone. |
| Authentication Level | Packet | Authenticates credentials and verifies that all call data received is from the expected client. |
| Impersonation Level | Impersonate | The server can impersonate the client while acting on its behalf, but with restrictions. The server can access resources on the same computer as the client. If the server is on the same computer as the client, it can access network resources as the client. If the server is a computer different from the client, it can access only resources that are on the same computer as the server. |

**Note:** When the Server starts, it will set the security level with the following call:

```
CSecurityDescriptor cSecurity;
cSecurity.InitializeFromThreadToken( );
CoInitializeSecurity(
cSecurity,
-1,
NULL,
NULL,
RPC_C_AUTHN_LEVEL_PKT,
RPC_C_IMP_LEVEL_IMPERSONATE,
NULL,
EOAC_NONE,
NULL);
```

### *Custom*

If neither of the preconfigured settings are suitable for your installation, you must choose *Custom*. When the selection is *Custom*, the server does not override the default security values. Instead, the settings you edited with DCOMCNFG are used.

### *Launch DCOMCNFG...*

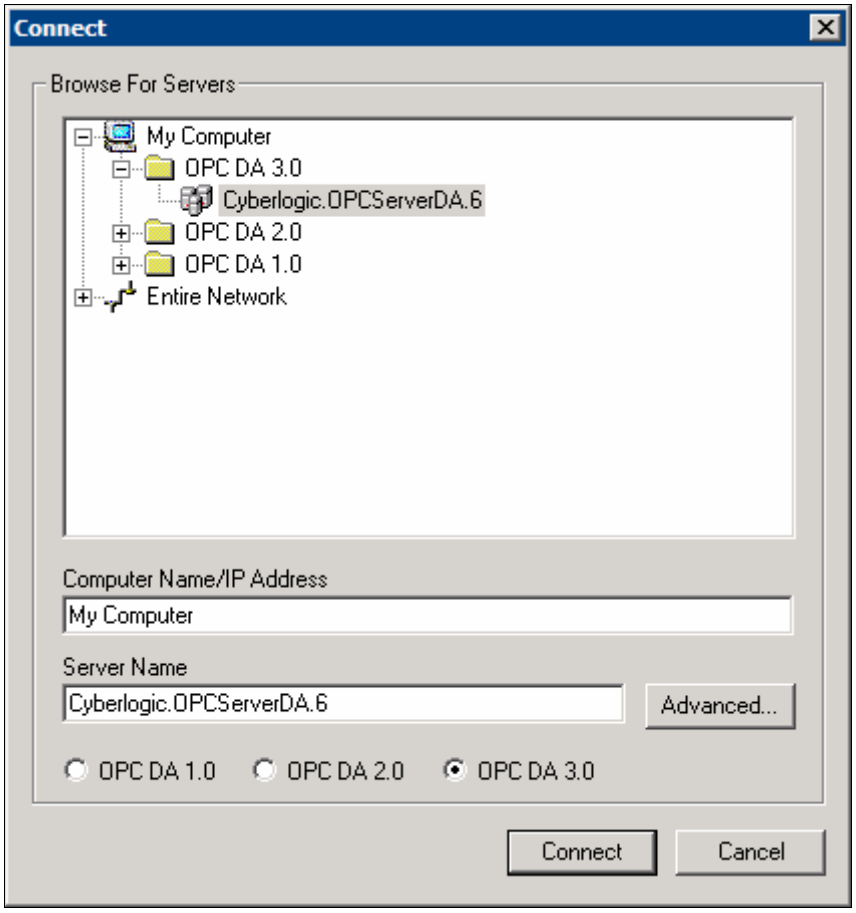
Click this button to configure the security settings manually.

If you selected *Custom*, you must use DCOMCNFG to configure all of the security settings. If you selected *Low* or *Medium*, you must use DCOMCNFG to configure all of the security settings except for the Cyberlogic OPC Server's access permissions, authentication level and impersonation level.

# OPC Client Connection

After completing your configuration, you will use an OPC client application to access the data from the Cyberlogic OPC Server. To do this, you must connect the client to the server. The exact method for doing this will vary somewhat from one client to another, but typically will be done by browsing a tree for available servers.

The figure below illustrates this process for the Cyberlogic OPC Client application. You would navigate through the tree, looking for the available servers on the local machine or on other computers attached to your network. When you find the desired server, select it and click *Connect*.




The servers available will be listed by their ProgID, Version-Independent ProgID or both. The example above shows a ProgID, with the *.6* at the end specifying that this is version 6 of Cyberlogic's OPC Server. A Version-Independent ProgID would be *Cyberlogic.OPCServerDA*, with no version number appended. In general, it is preferable to use the Version-Independent ProgID to avoid the need to change the selection if you install a newer version of the server software. Some servers may allow you to have multiple versions installed at the same time. In such a case, you would use the ProgID to specify the exact version to connect to.

To connect an OPC Alarms and Events client, you would proceed in the same manner, selecting among the available AE servers. In the case of the Cyberlogic Alarms and Events server, the ProgID would be *Cyberlogic.OPCServerAE.6*. The procedure for connecting an XML Data Access client application is explained in Appendix D: OPC XML Data Access Support.

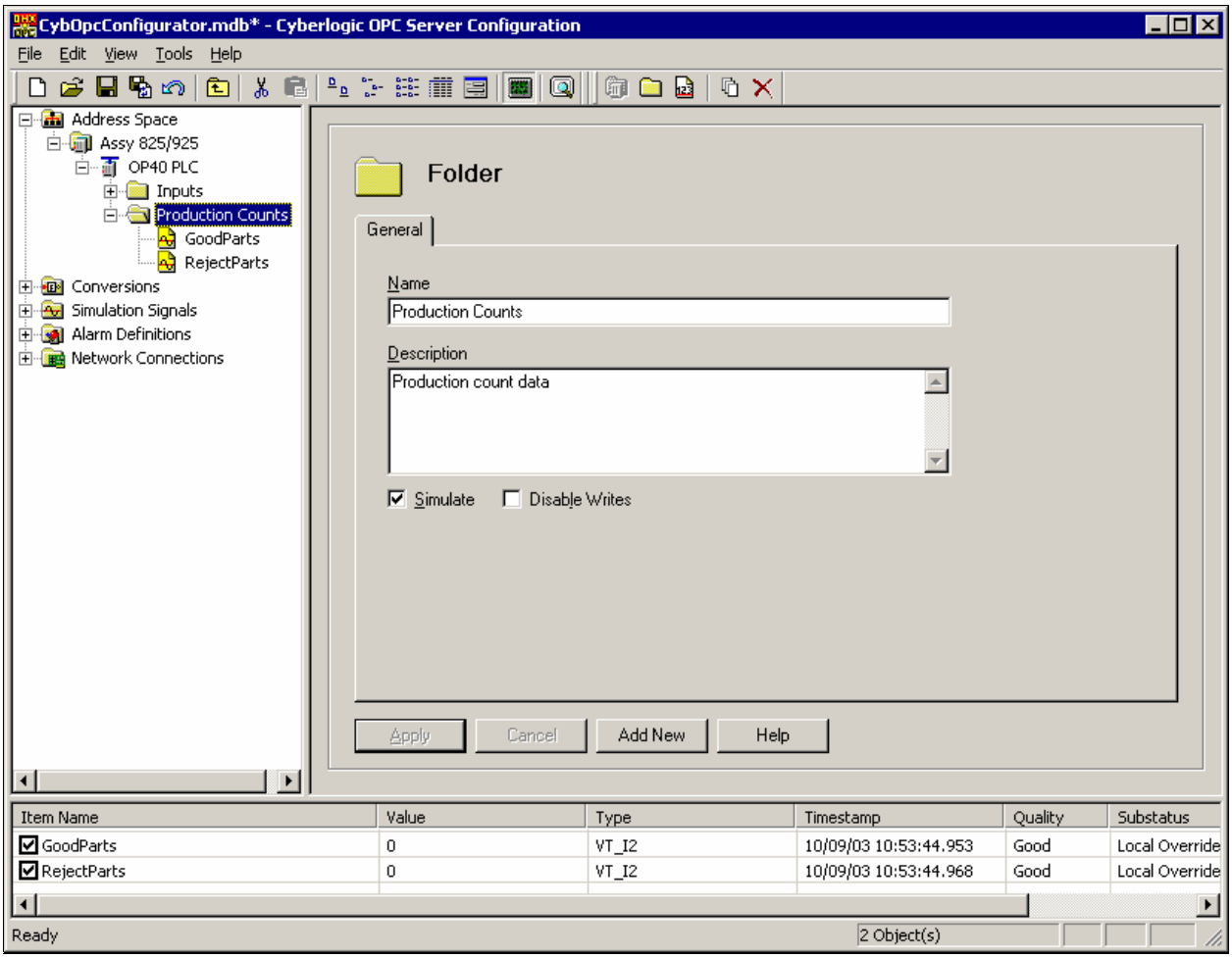
# VALIDATION & TROUBLESHOOTING

The following sections describe features that will help you to verify and troubleshoot your Server's operation. The Data Monitor and Cyberlogic OPC Client allow you to view the data as it is received by the Server. Microsoft's Performance Monitor allows you to view relevant performance information. The DirectAccess feature lets you look at data values even if they have not been configured as Data Items. The Event Viewer may provide important status or error messages. Finally, there is a list of Cyberlogic OPC Server Messages and Frequently Asked Questions to assist in your troubleshooting.

## Data Monitor

The Cyberlogic OPC Server Configuration Editor includes a built-in utility called the Data Monitor. It is a diagnostic tool that allows you to monitor the values of the Data Items. You can enable the Data Monitor from the View menu or by right-clicking and selecting *Data Monitor* from the context menu.

With the Data Monitor enabled, the Cyberlogic OPC Server Configuration Editor acts as an OPC client to the Cyberlogic OPC Server. It creates an output display pane at the bottom of the main workspace window.




Each row in the Data Monitor corresponds to a Data Item in the selected Data Item folder.

### *Enable Checkbox*

At the left end of each Data Item row is a checkbox that, when checked, enables monitoring of its Data Item. By default, this checkbox is not checked. To minimize unnecessary communications, enable only Data Items that you are interested in.

### *Item Name*

Shows the name of the monitored Data Item.

### *Value*

The Data Item's current value.

### *Type*

The Data Monitor always requests data in the canonical (native) format. Therefore, the Type column shows the canonical data type of the requested data.

### *Timestamp*

The timestamp for the Data Item's current value.

### *Quality, Substatus and Limit Status*

Each data value returned by an OPC server has a 16-bit quality flag word associated with it. The low eight bits are currently defined in the form of three bit fields: Quality, Substatus and Limit Status. The Data Monitor displays the current value of each of these fields for the Data Item value. For more information, refer to the OPC Quality Flags section in the OPC Data Access specification.

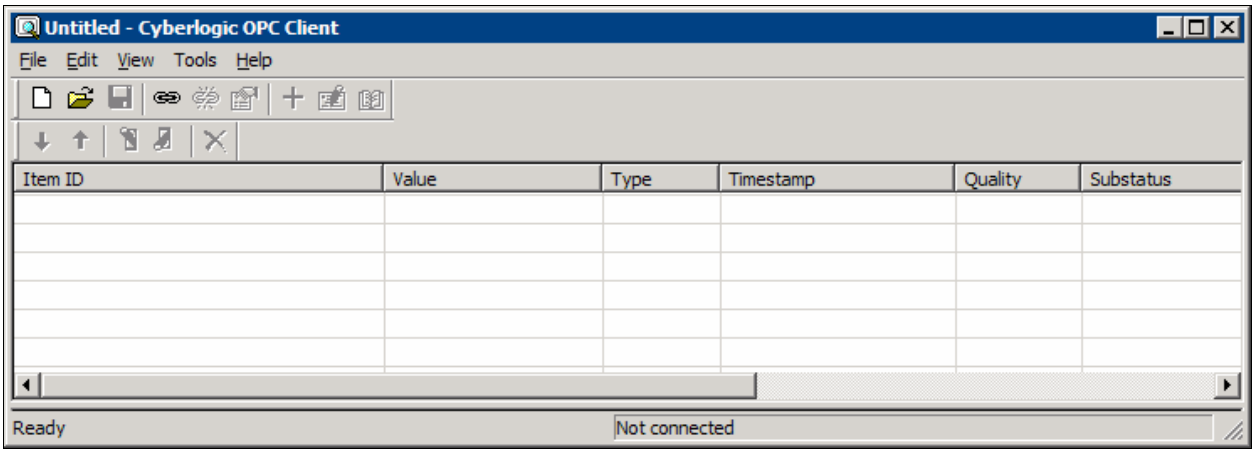
# Cyberlogic OPC Client

The Cyberlogic OPC Client is a stand-alone OPC DA client that you can use to test the operation of the Cyberlogic OPC Server and other OPC servers. Although its appearance resembles the Cyberlogic OPC Server Data Monitor, it contains many additional features. An important feature is its ability to view real-time values of Data Items from more than one device simultaneously. Running multiple instances of the Client can also be useful when testing the Server's response to various loads.

## *Typical Client Session*

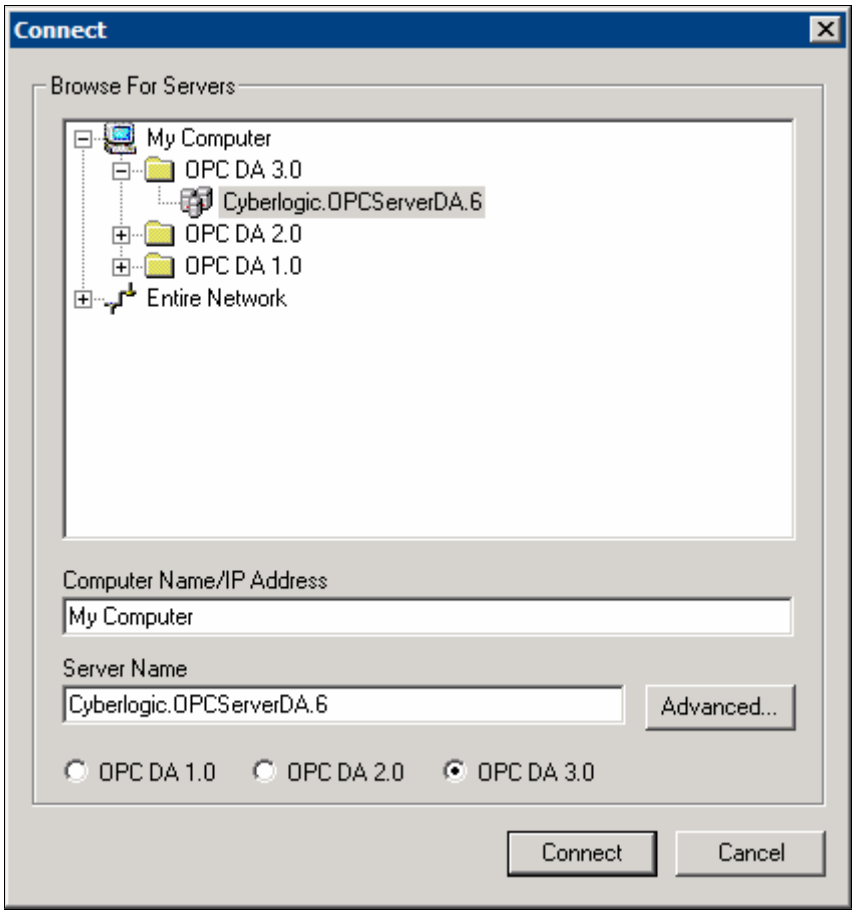
The following steps show how to use the Cyberlogic OPC Client. Use this only as a guideline. Only the most common features are shown here. A detailed description of each feature follows this demonstration section.

1. Select *OPC Client* from the OPC Server group on the Windows Start menu. You will see the following screen:



2. Click the *Connect* button or select *Connect…* from the *File* menu. The client application can connect to OPC servers running locally or on other machines on a network. Notice that the servers are organized into folders according to the OPC Data Access spec level they support. Double-click on the server to which the Client should connect.

   You will normally want to select the server *Cyberlogic.OPCServerDA.6* from the OPC DA 3.0 folder.

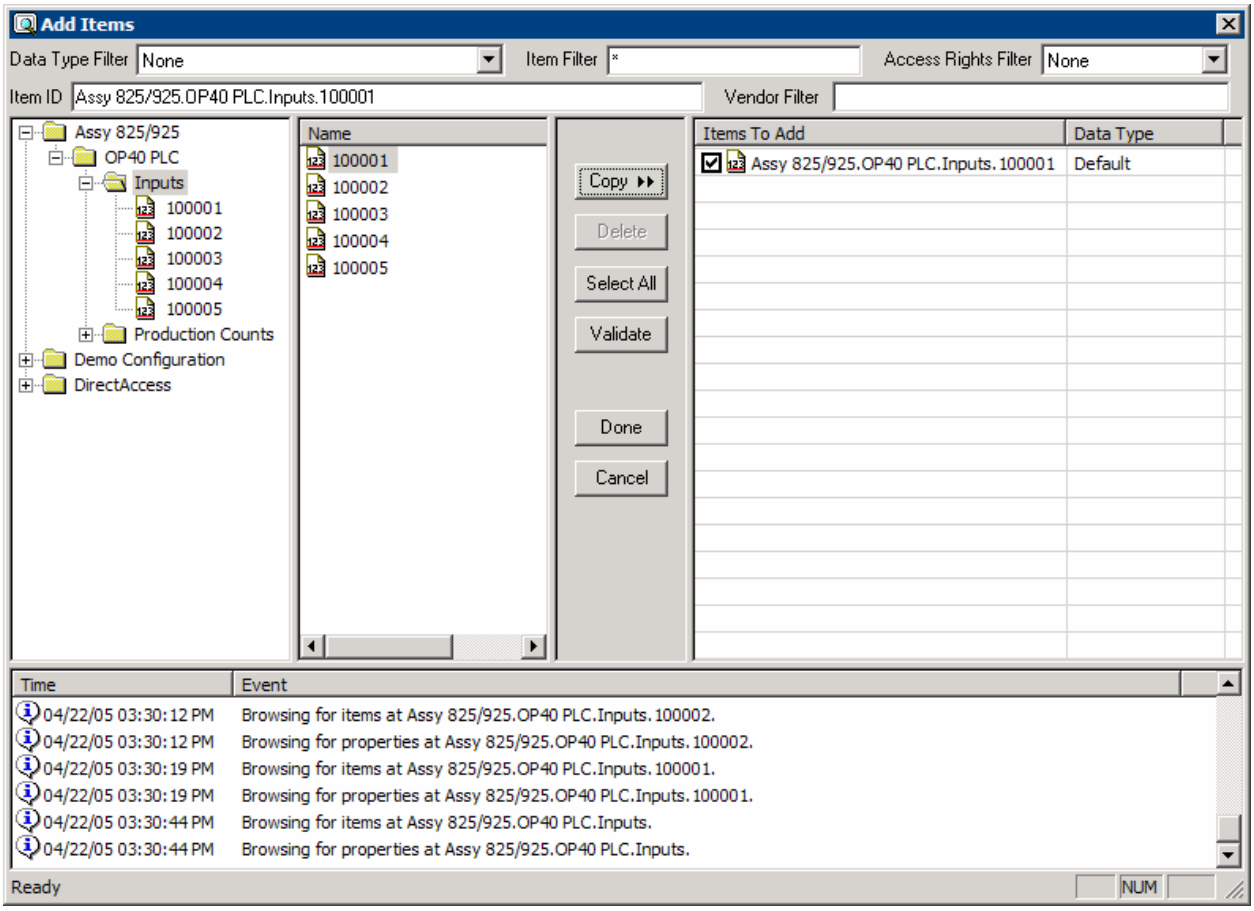



   The radio button selection allows you to restrict the client's operation to DA 1.0a level interfaces, 2.05a and below or 3.00 and below.

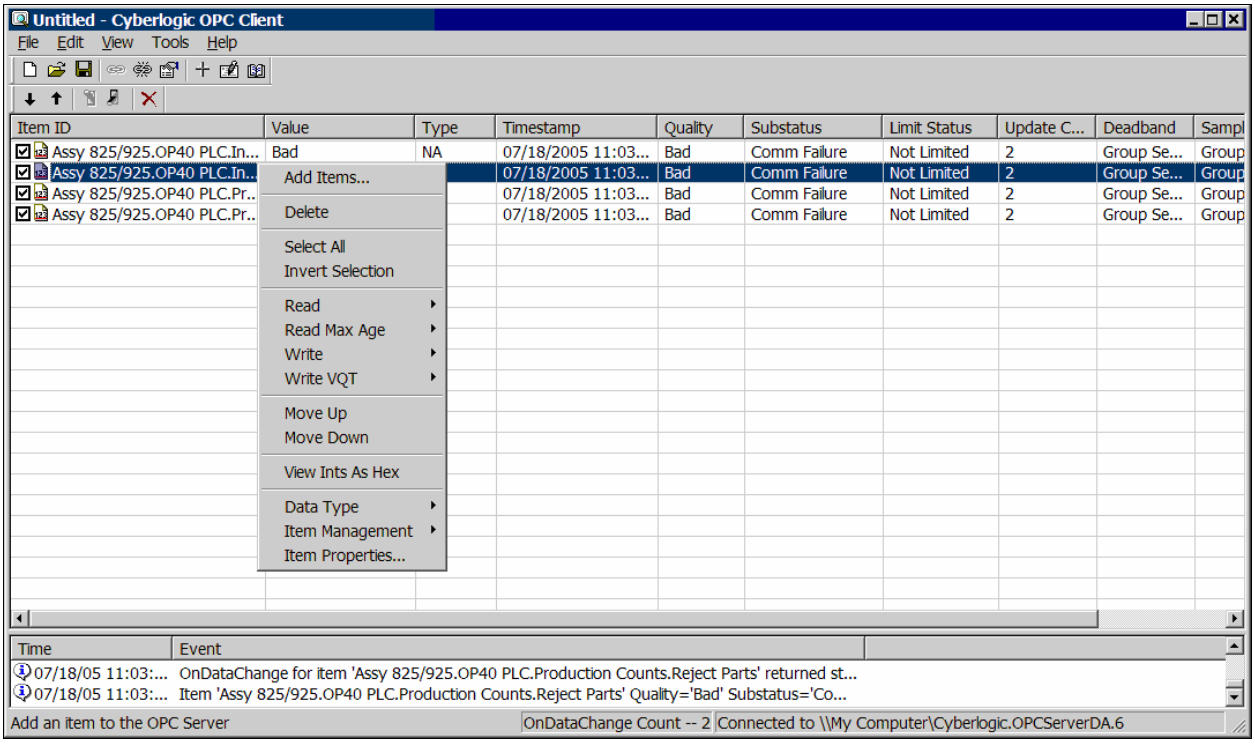
3. Next, add Data Items to the list by clicking the *New Item* button,  , or by selecting *New Item…* from the Edit menu. Navigate through the tree to find the folder containing the Data Item you wish to display. Select the Data Item from the Name box and click *Copy* to add it to the list. If you instead select the Data Item from the tree, the Name box will display the properties of that item that can be displayed.

You may limit the items shown in the Name box by using the Data Type Filter, Item Filter and Access Rights Filter boxes. Once an item is on the Items To Add list, you can uncheck it to disable updates.

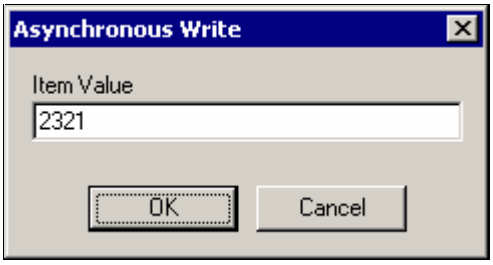
When all of the desired Data Items are selected, click *Done*.



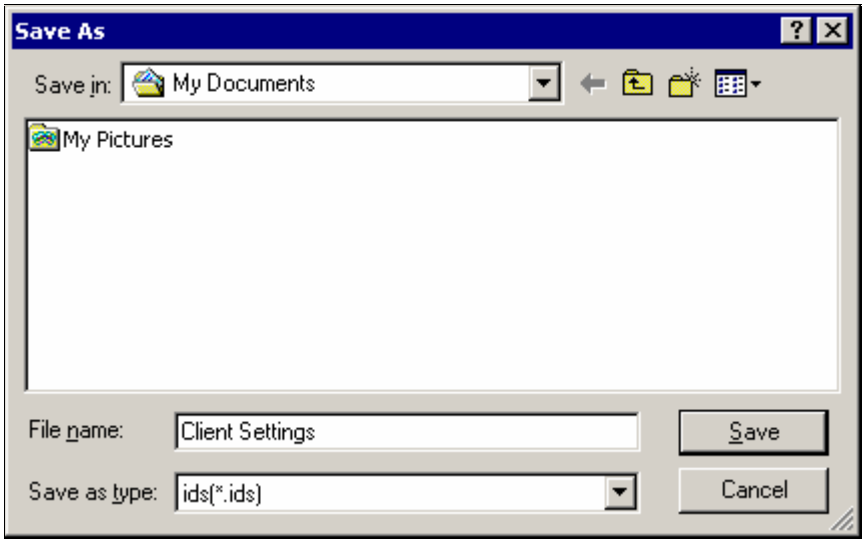
4. A Data Item must be enabled for the client to display its real-time values. To enable an item, check the box next to the item or select the item and either click the *Enable* button or select *Enable* from the Edit menu. Right-click on an item to open a context menu that allows you to delete the item, move it up or down the list, view its properties and do other useful functions.




5. The Cyberlogic OPC Client allows you to write values to Data Items that are not write-protected by clicking the toolbar's *Asynchronous Write* button,  , or by selecting *Write / Asynchronous…* or *Write / Synchronous…* from the Edit menu.



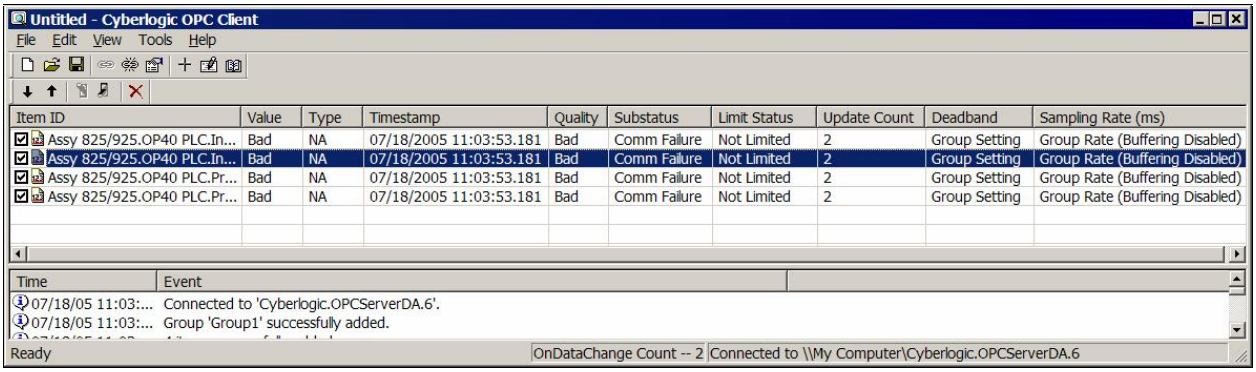
6. You can save the Client configuration to a disk file by choosing *Save As…* or *Save…* from the File menu. To reload a previously saved configuration, select *Open…* from the File menu and then choose the *.ids* file with the desired configuration.




7. To disconnect from the Server, click the *Disconnect* button or choose *Disconnect* from the File menu.

# *Main View Window*

The main window of the Cyberlogic OPC Client resembles the Data Monitor of the Cyberlogic OPC Server Configuration Editor. Each Data Item uses one row with several data fields.




## *Enable Checkbox*

At the left end of each Data Item row is a checkbox that, when checked, enables monitoring of its Data Item. By default, this checkbox is not checked. To minimize unnecessary communications, enable only Data Items that you are interested in.

## *Item ID*

This is the fully qualified Item ID string for the Data Item. The OPC client must use this string to access the Data Item.

## *Value*

The Data Item's current value.

## *Type*

This column shows the data type of the requested data. You may change this by right-clicking on a Data Item and selecting Data Type from the context menu.

## *Timestamp*

The timestamp for the Data Item's current value.

## *Quality, Substatus and Limit Status*

Each data value returned by an OPC server has a 16-bit quality flag word associated with it. The low eight bits are currently defined in the form of three bit fields: Quality, Substatus and Limit Status. The Data Monitor displays the current value of each of these fields for the Data Item value. For more information, refer to the OPC Quality Flags section in the OPC Data Access specification.

### *Update Count*

This is simply the number of times the Data Item has been updated.

### *Deadband*

This applies to DA 3.0 analog values only. It is the minimum change in a value, expressed as percent of full scale, that must be exceeded for the data value to be updated in the cache on the basis of the value having changed.

### *Sampling Rate*

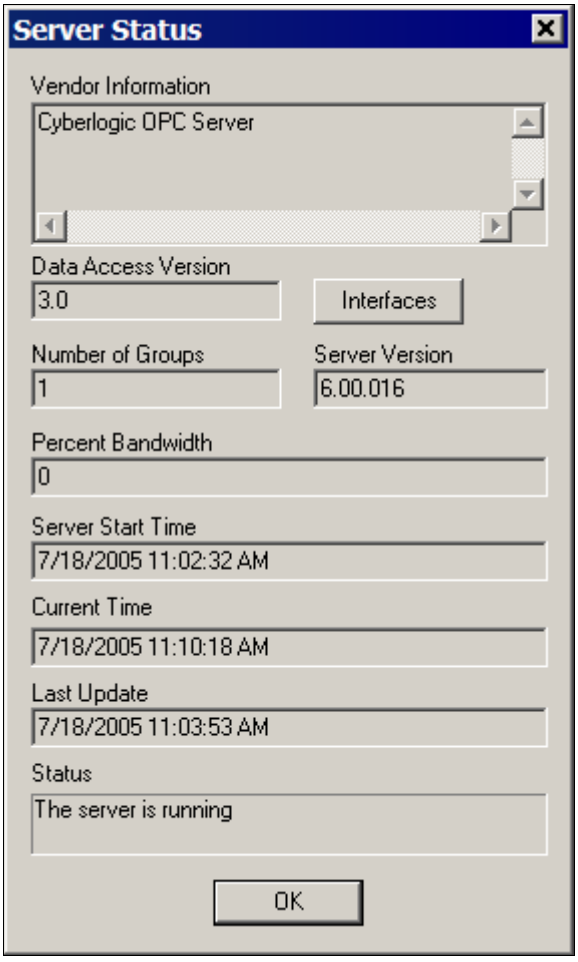
This applies to DA 3.0 values only. It is the interval at which the Server will read the item's value from the device.

## Arranging Data Item Order

By default, the Client controls always add a new Data Item at the end of the Data Item list. To change this order, select an item and then click the *Move Up* or *Move Down* button on the Edit Toolbar (or select the *Move Up*/*Move Down* from the Edit menu). You can sort the Data items in ascending or descending order based upon values in each column by clicking on the column's header bar.

# *Server Status*

To view the Server properties, select *Server Status…* from the File menu. The display below will appear.




## *Data Access Version*

The Data Access Version indicates the OPC Data Access specification level used by the client to communicate to the server.

## *Interfaces*

This button is available only for DA 3.0 servers. Click it to display a list of all of the COM interfaces supported by this server object.

## *Number of Groups*

This indicates the total number of groups being managed by the server on behalf of this client. Since the Cyberlogic OPC Client uses a single group for all data items, this value is always equal to 1.

### *Server Version*

This field displays the software revision level of the OPC Server.

### *Percent Bandwidth*

The behavior of this field is server-specific. Typically, it shows the approximate percent of bandwidth currently used by server. A value greater than 100% indicates that the combination of items and update rate is too high. The Server may also return 0xFFFFFFFF if this value is unknown.

### *Server Start Time*

This is the time (UTC) that the Server was started. This is constant for the Server instance and is not reset when the Server changes states.

### *Last Update*

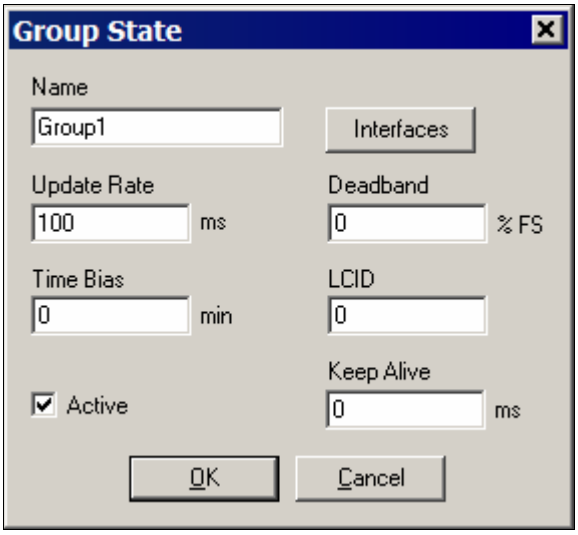
This is the time the server last sent a data value update.

### *Status*

This text indicates the status of the server.

# *Group State*

All of the Data Items you are viewing are part of a single OPC group. The Group State… option of the Edit menu allows you to view and edit various parameters for this group.




### *Name*

This is simply the name of the group.

### *Interfaces*

This button is available only for DA 3.0 servers. Click it to display a list of all of the COM interfaces supported by this group object.

### *Update Rate*

This is the update rate returned by the Server, which may be different from the requested update rate. It is actually the interval between updates, specified in milliseconds. In general, the OPC Server rounds the requested value up to the next available supported rate.

### *Deadband*

The percent change in an item value that will cause the Server to send an update to a client. This parameter only applies to analog items. The range of the Deadband is from 0 to 100 percent of full scale (FS).

### *Time Bias*

The Time Bias indicates the time zone in which the data was collected and is specified in minutes. The data collection time zone may be different from the time zone of either the client or the server. The default Time Bias for the group is that of the server.

| | |
|---|---|
| **Note:** | The Time Bias behaves like the Bias field in the Win32 TIME_ZONE_INFORMATION structure, which is to say it does not account for daylight savings time (DST). |

The Time Bias is set only when the group is created or when Set State is called. In general, a Client computes the data's local time by: Time Stamp + Time Bias + DST Bias (if any).

### *LCID*

The *LCID* identifies the language the Server uses when returning values as text. The following table shows the LCID codes for a few common languages:

| Language | LCID (hex) |
|---|---|
| English (United States) | 0x0409 |
| German (Standard) | 0x0407 |
| French (Standard) | 0x040c |
| Spanish (Traditional) | 0x040a |
| Italian (Standard) | 0x0410 |

A zero value indicates that the local language should be used.

### *Active*

Groups and Items within Groups have an Active Flag. If you clear the *Active* checkbox for a group, you will disable all Data Item updates for that group. The active state of the group is maintained separately from the active state of the individual Data Items, so changing the state of the group does not change the state of the items.

### *Keep Alive*

This feature is available with DA 3.0 servers only. When a subscription has a non-zero Keep Alive time, the client will receive a callback on the subscription at least at the rate indicated by the Keep Alive time, even if there are no new events to report.

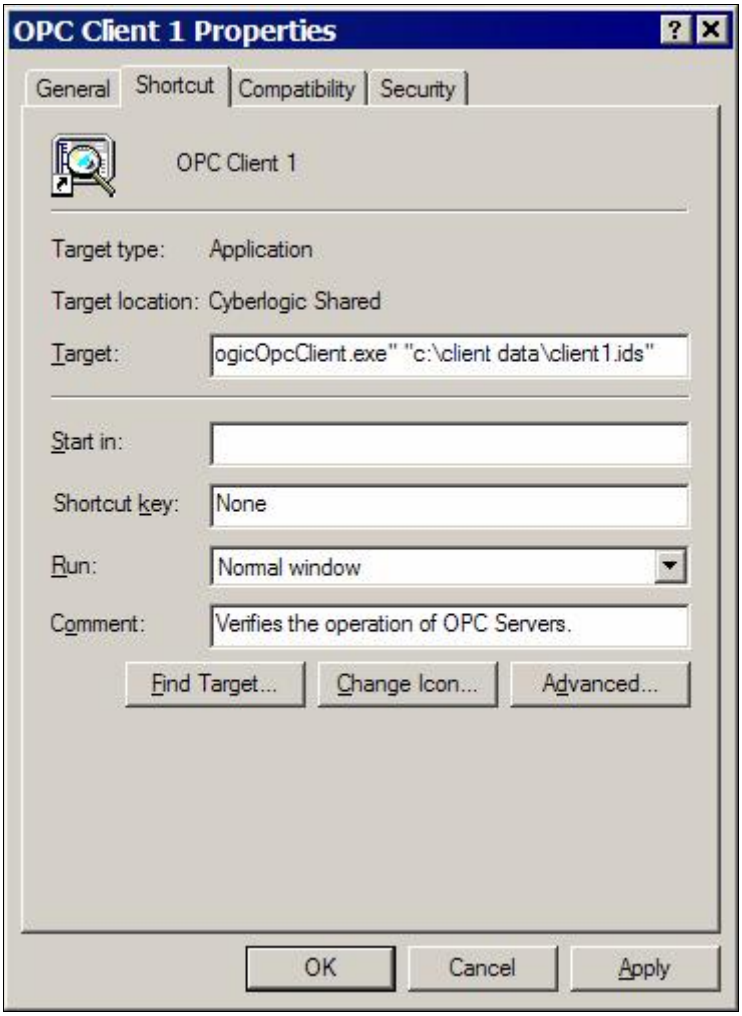
## *Saving/Reloading Data Items*

To speed up the process of entering Data Items, the current client configuration can be saved to an ids file. To do that, choose *Save…* from the File menu. You can reload a previously saved configuration by selecting *Open…* from the File menu and then choosing the ids file with the desired configuration. For either of these options to be enabled, the client must be connected to the server. In addition, the items contained in the ids file must correspond to those items available within the present server configuration.

# *Shortcuts and Command Line Parameters*

For quick access to client configuration files you have saved, you can set up Windows shortcuts that will open the client and load a specified ids configuration file. To do this, open the Windows *Start* menu and navigate to the sub-menu for the OPC Server product you have installed. Right-click on the *OPC Client* entry, drag it to the desktop, then drop it and select *Copy Here* from the pop-up menu.

Right-click on the shortcut and select *Properties* from the context menu. The shortcut properties box, below, will open. On the *Shortcut* tab, add the path and configuration file name to the *Target* field. Notice that the path and file name must be enclosed in quotes and preceded by a space. Click *OK* to save the changes. Now, when you double-click the shortcut, the client will open and automatically load the configuration file.



# *Client Options*

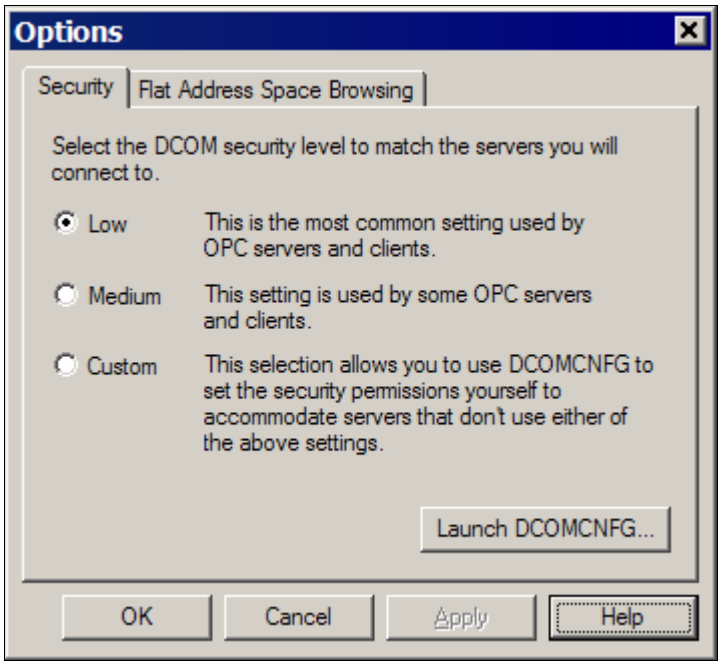
The *Options…* entry in the *Tools* menu allows you to set the client property preferences.

## Security Tab

For an OPC client and server to communicate, both systems must have proper DCOM security settings. There is no standard setting for OPC applications because each installation's security needs are unique. Therefore, you must decide how to configure security for your systems. For help in making these decisions and a detailed discussion of how to do the configuration, refer to the document *OPC & DCOM: A Guide to Using the Cyberlogic OPC Server via DCOM*. A copy of this document was installed on your system along with the software, and can be accessed from the Windows Start menu by navigating to the OPC Server directory and clicking on *DCOM Help*.

As part of the security configuration, you must select the access permissions, authentication level and impersonation level to be used. Two settings for these parameters are used as defaults by many OPC-based products. These may be selected by choosing the *Low* or *Medium* settings from this screen. If neither of these is the correct setting for your situation, you must choose *Custom* and click *Launch DCOMCNFG...* to configure all of the security settings yourself through the Windows operating system.

| | |
|---|---|
| **Caution:** | The preconfigured *Low* and *Medium* security settings override only the access permissions, authentication level and impersonation level for the Cyberlogic OPC Client. The rest of the security settings configured with the DCOMCNFG utility still apply. |




| | |
|---|---|
| **Caution:** | If you change the selection on this tab, you must close the client and reopen it before the new settings will take effect. |

If the *Low* or *Medium* security settings match your server's settings, select the appropriate radio option and click the *OK* button.

### *Low and Medium*

For details of these settings, refer to the discussion of the Server Options Security Tab.

### *Custom*

If neither of the preconfigured settings are suitable for your installation, you must choose *Custom*. When the selection is *Custom*, the client does not override the default security values. Instead, the settings you edited with DCOMCNFG are used.

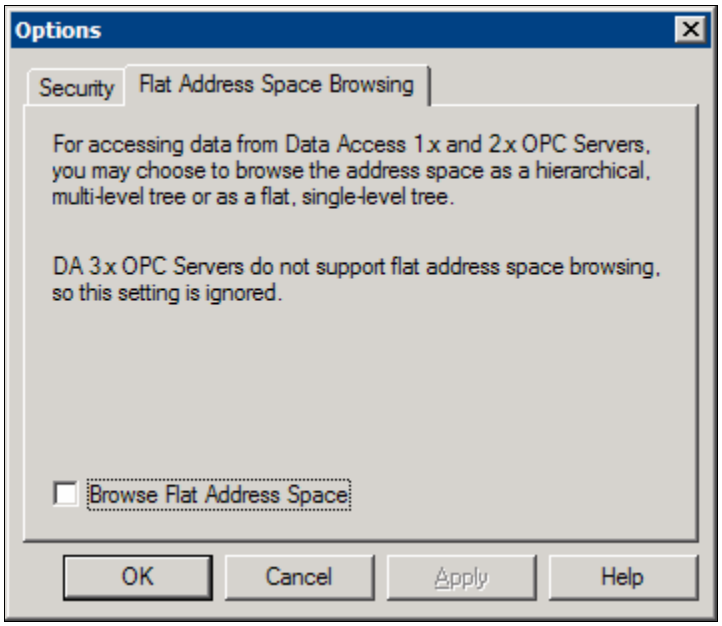
### *Launch DCOMCNFG...*

Click this button to configure the security settings manually.

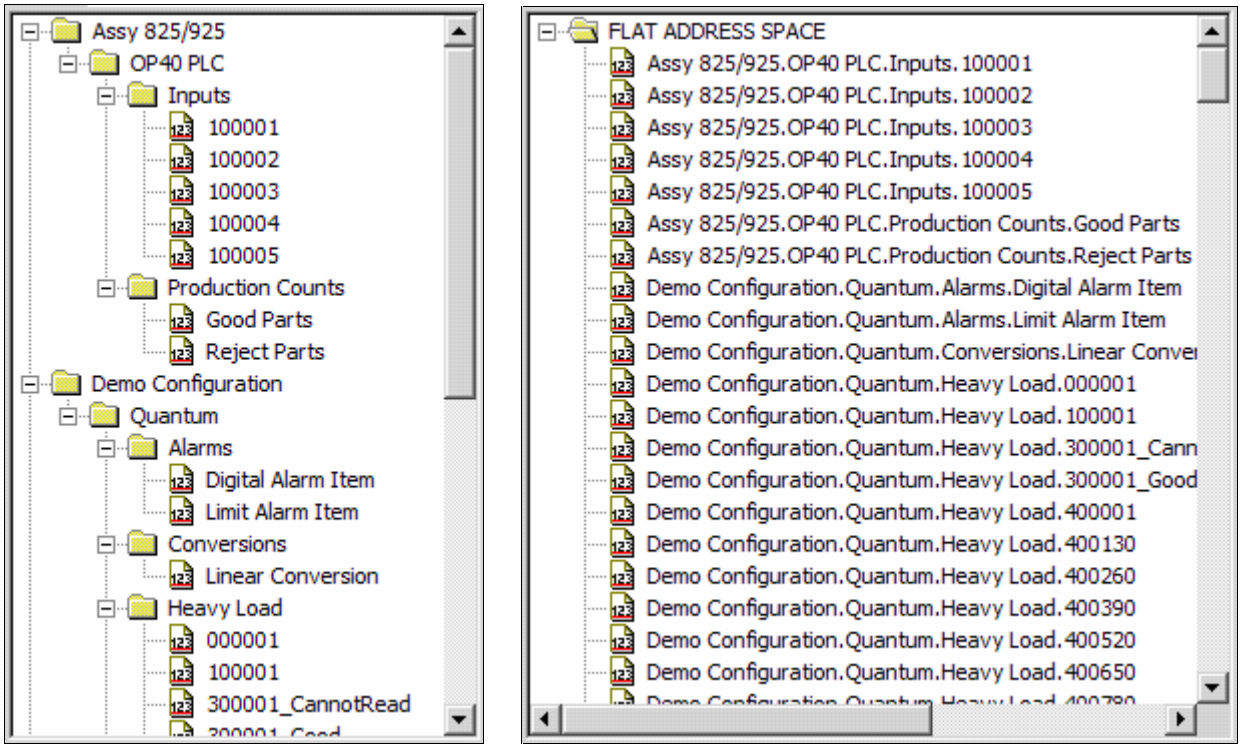
If you selected *Custom*, you must use DCOMCNFG to configure all of the security settings. If you selected *Low* or *Medium*, you must use DCOMCNFG to configure all of the security settings except for the Cyberlogic OPC Client's access permissions, authentication level and impersonation level.

## Flat Address Space Browsing Tab

The OPC Server address space is normally shown with a multi-level tree structure. When browsing for elements, you may prefer to see the address space as a flat structure with all of the elements at a single level. This is supported in Data Access 1.x and 2.x compliant servers, but not in 3.x compliant servers. If the server you are using supports this feature, you can use it by checking *Browse Flat Address Space*.




The figures below show the difference between the two views.




*Flat Address Space Browsing:*
*The right pane shows the normal view and the left pane shows the flat view.*

# Performance Monitor

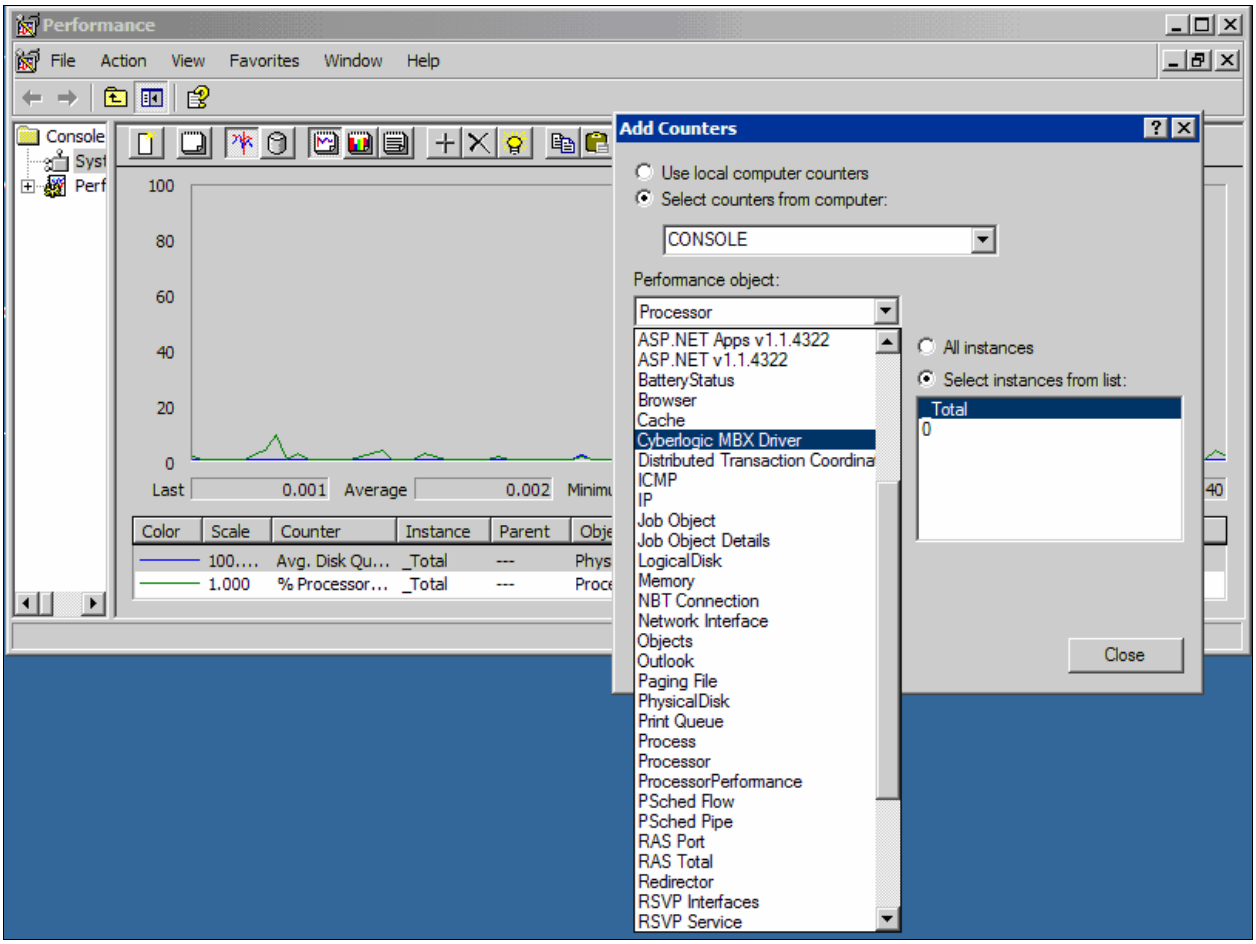
Microsoft provides the Performance Monitor diagnostic tool as part of the Windows XP/2000/NT operating system. Applications that support it, including the Cyberlogic drivers, allow users to monitor relevant performance information. Multiple devices can be monitored simultaneously for comparison.

To start this program, click on its icon from Start/Programs/Administrative Tools group.

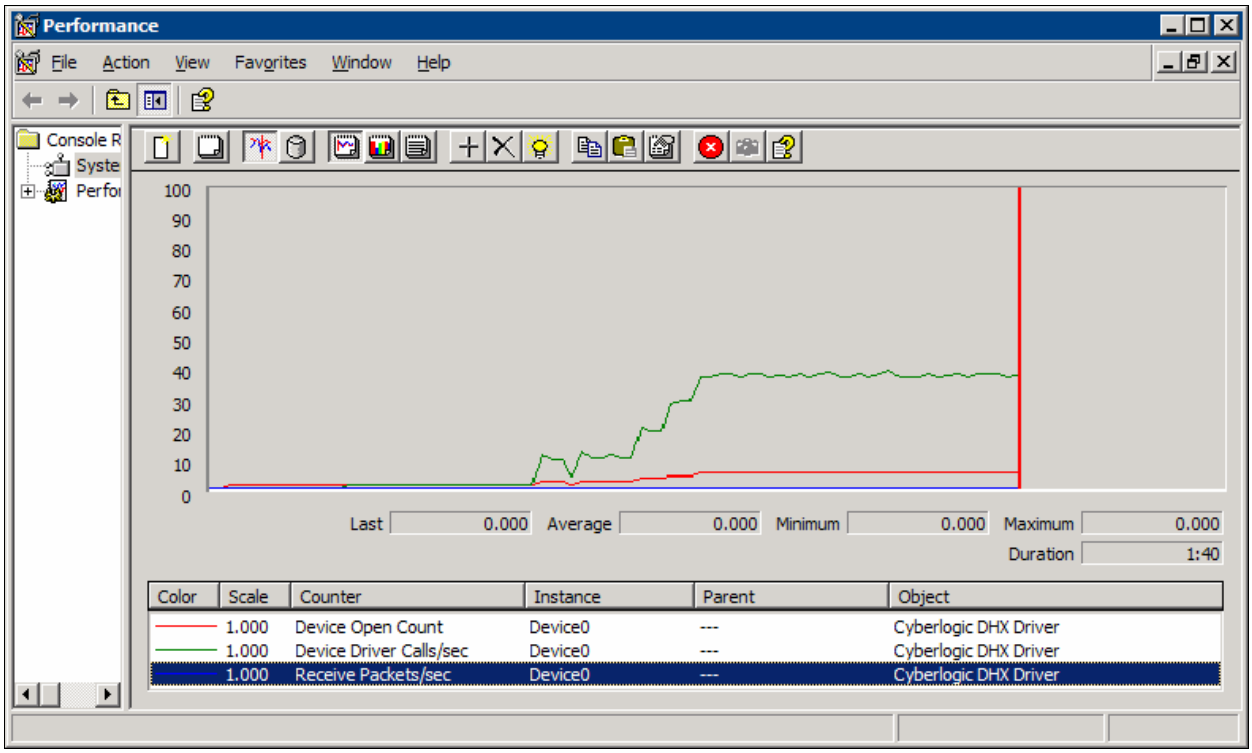
## How to Use the Performance Monitor

Since extensive help is provided for this program by Microsoft, only a few points that are relevant to the Cyberlogic drivers will be shown here. In this example, we will use the tool to check the performance of the DHX Driver. You would use the same procedure for any of the other drivers.

When the Performance Monitor program starts, select the *Add to Chart* option from the Edit menu (or click the + button on the tool bar) and select *Cyberlogic DHX Driver* from the Object list. After choosing a monitoring option, click *Add* and then *Done*.



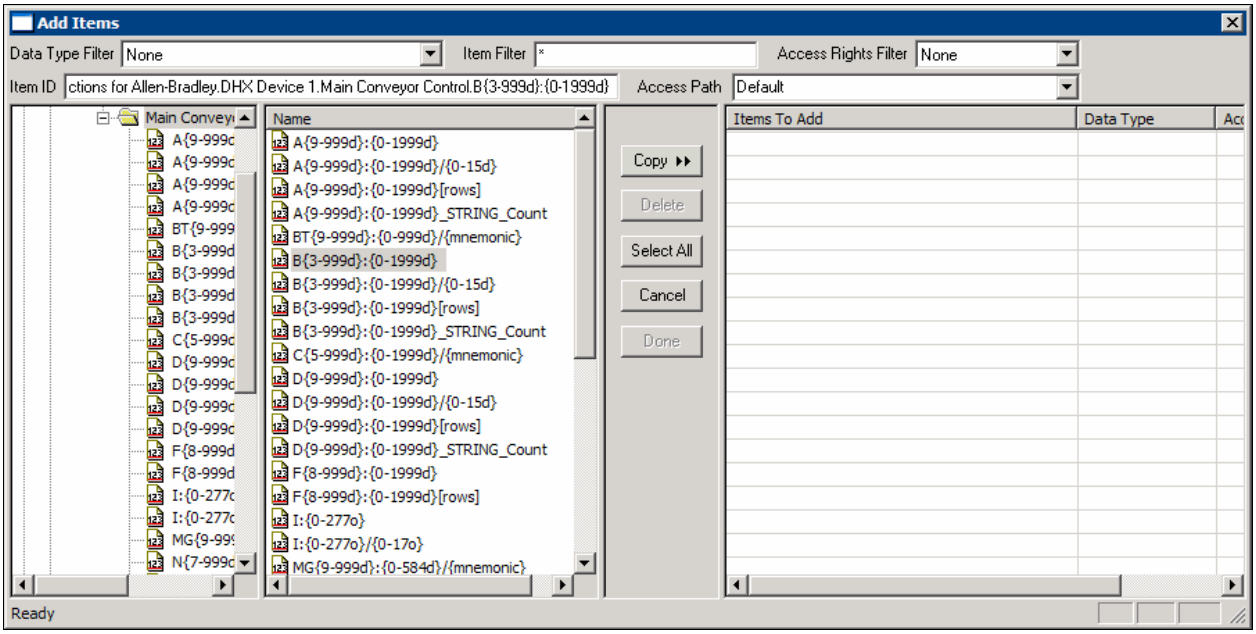
Three of the many monitoring options are shown below.



# DirectAccess

At run time, in addition to the user-configured branches, the Cyberlogic OPC Server dynamically creates a branch called DirectAccess at the root of its Address Space. The DirectAccess branch acts like a Device Folder that contains all of the configured Network Connections. Each Network Connection branch contains its configured Network Nodes. However, only Network Nodes that enable DirectAccess are present. OPC clients can then use this branch to access any register in any configured Network Node by directly specifying the register address.

For Network Nodes in the DirectAccess branch, the Cyberlogic OPC Server reports a list of hints about the types of Data Items that may exist on the selected node. These are not valid item addresses. Rather, they are just hints for the user to help you specify a proper address. The example below is for a PLC-5.




In this example, B{3-999d}:{0-1999d} is an address hint. The *B* indicates the file type, which in this case is a binary file. The next field — *{3-999d}* — specifies the file number, which must be a decimal number between 3 and 999. A colon follows the file number. The last field — *{0-1999d}* — specifies the register number, which must be a decimal number between 0 and 1999.

Therefore, to access the register located at B3:100, you would edit the Item ID field to read:

*DirectAccess.DHX Connections for Allen-Bradley.DHX Device 1.Main Conveyor Control.B3:100*

An input address hint might be of the form I:{0-277o}/{0-17o}. In this case, the number ranges are in octal and a typical address is I:3/1.

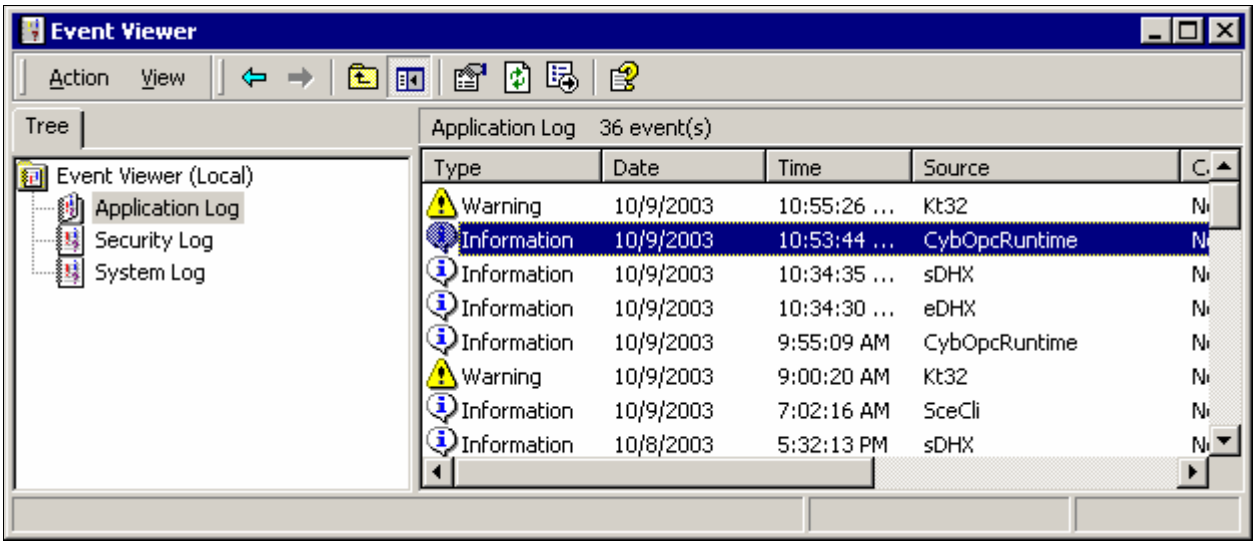
# Event Viewer

The Cyberlogic OPC Server may detect a run-time error. When it does, the Server sends an appropriate message to the Windows XP/2000/NT Event Logger. You can then view these messages using the following procedure:

1. From the Control Panel/Administrative Tools program group, run *Event Viewer*.

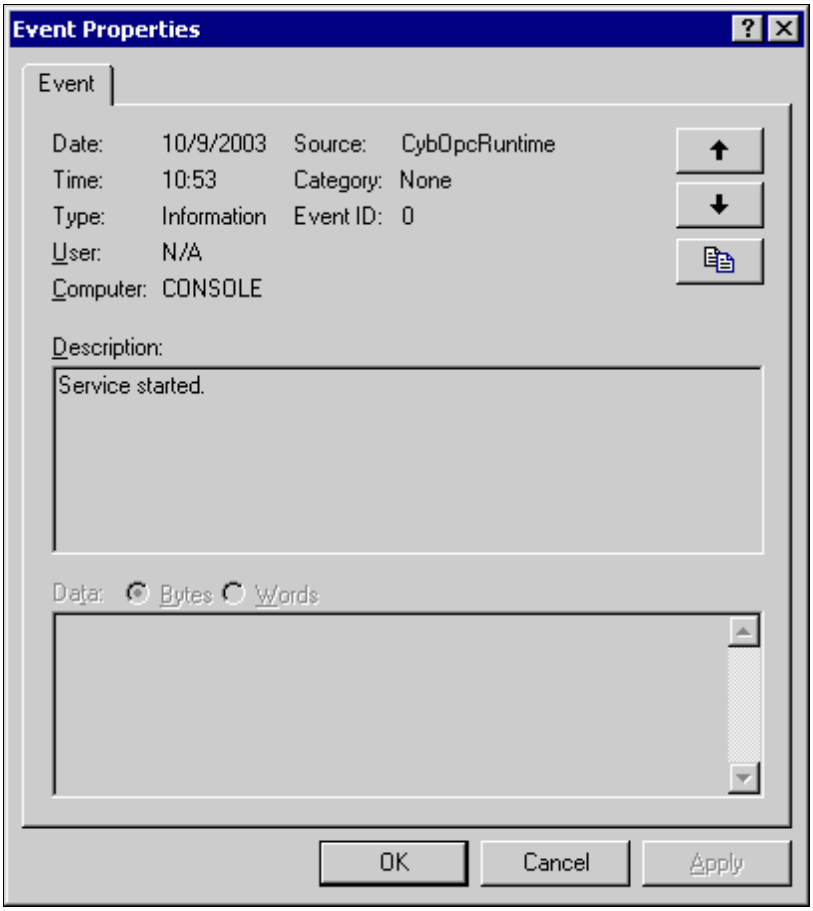
2. Select *Applications* from the Log menu or from the tree in the left side menu, depending upon your operating system.

3. Look for entries with *CybOpcRuntime* in the Source column.




| | |
|---|---|
| **Caution:** | The Event Viewer does not clear itself after the system reboots. Check the time stamps of the messages to be sure you are not looking at an old error message. |

4. Double-click on the selected entry to display the complete event message as seen below.




5. For further descriptions of the error log messages, refer to the Cyberlogic OPC Server Messages section.

# Cyberlogic OPC Server Messages

This section shows Error Log messages that can be generated by the main Cyberlogic OPC Server module. Each driver agent can also log error messages. For a list of these messages, refer to the help file for the driver agent you are using.

## Errors

***Memory allocation error in <function name>. Close some applications. Add more memory to your system. Contact the manufacturer's technical support.***

The specified function failed to allocate the needed memory. This is a fatal error. If you are running low on memory, close some applications or add more memory to your system. If the problem continues, contact technical support for more information on a possible solution.

***Unexpected error in <function name>. Please contact the manufacturer's technical support.***

Indicates a possible programming bug in the Server. Contact technical support for more information on a possible solution.

***Unexpected error in <function name> (Error code = <error code>). Please contact the manufacturer's technical support.***

Indicates a possible programming bug in the Server. Contact technical support for more information on a possible solution.

***Handler not installed.***

During startup, the Cyberlogic OPC Server failed to register the Service Control Handler. Contact technical support for more information on a possible solution.

***Bad service request.***

The Cyberlogic OPC Server detected an unsupported service request. Contact technical support for more information on a possible solution.

## Warnings

***Memory allocation error in <function name>. The server may not operate correctly. Close some applications. Add more memory to your system. Contact the manufacturer's technical support.***

The specified function failed to allocate the needed memory. The server will continue to operate, but some functions may not work. If you are running low on memory, close some applications or add more memory to your system. If the problem continues, contact technical support for more information on a possible solution.

***Communication module for driver agent <driver agent name> failed to load (Error code = <error code>). All configuration related to this driver agent will be ignored. To resolve this problem, please contact the manufacturer's technical support.***

It you believe that the indicated driver agent is installed, this message may indicate that you have corrupted installation. Refer to the *Repairing the Cyberlogic OPC Server Installation* section for information on repairing corrupted installation. This error will also be generated if your configuration file was generated on a system with the listed driver agent installed, but not installed on your system. Install the listed driver agent to correct the problem.

## Informational

***Service started.***

The Cyberlogic OPC Server started successfully.

***Service stopped.***

The Cyberlogic OPC Server has stopped.

# Frequently Asked Questions

***I am not connected to the network, but my data items are being updated. What's going on?***

Simulation Signals are generating data for the items. Verify that the Simulate check box is cleared for the Data Item and for each parent up the chain from the Data Item.

***I created a new data item in the Address Space. When I try to look at it with the Data Monitor, the item is shown as Not Available. The entire configuration appears correct. What do I do?***

You may not have updated the runtime module of the Cyberlogic OPC Server with your recent configuration changes. To save current configuration and update the server, select *Save & Update Server* from the File menu of the OPC Server Configuration Editor (or click the *Save & Update Server* button on the standard buttons toolbar).

***I can't find any information specific to my communication network in your help file.***

This help file describes only the common features of the Cyberlogic OPC Server. For information related to a particular driver agent, refer to the help file specific for that agent.

***I cannot update the Server with my new or changed configuration. It always returns a message that the update failed.***

On some systems, Windows will spontaneously unregister the Server application. You must manually re-register it. To do this, locate the file *CybOpcRuntimeService.exe*. In most installations, it will be in the directory *C:\Program Files\Common Files\Cyberlogic Shared\OPC*.

From the Accessories group of the Windows Start menu, run *Command Prompt.* (In the following commands, <Enter> indicates that you should press the *Enter* key.)

You must change to the drive and directory you just located. To change the drive to C:, for example, type:

*C: <Enter>*

Once you are at the correct drive, change the directory by typing this command. (There is a space after the *cd*.)

*cd \Program Files\Common Files\Cyberlogic Shared\OPC <Enter>*

Now type the following. (Note that there is a space before the slash.)

*CybOpcRuntimeService.exe /unregserver <Enter>*

Finally, type the following. (Again, there is a space before the slash.)

*CybOpcRuntimeService.exe /service <Enter>*

Close the Command Prompt window.

***I cannot get XML Data Access to work. The Event Viewer shows an error for a User called ASPNET, with the description: "Access denied attempting to launch a DCOM Server using DefaultLaunchPermission".***

You must add the User account *ASPNET* to the DCOM security settings for the OPC Server software, giving it launch and access rights. To do this, you will use the DCOM configuration editor, dcomcnfg.exe. For details on how to use this editor, refer to the document *OPC & DCOM: A Guide to Using the Cyberlogic OPC Server via DCOM*. A copy of this document was installed on your system along with the software and can be accessed from the Windows Start menu by navigating to the OPC Server directory and clicking on *DCOM Help*.

Note that on some systems, you may not get the Event Viewer error message.

# APPENDIX A: ITEM PROPERTIES

In the Cyberlogic OPC Server, all Data Items have properties, or attributes, associated with them. These attributes are values related to the Data Item. The OPC Data Access specification defines several standard properties and allows vendors to define custom properties.

## Standard Properties

Properties with IDs from 1 - 4999 are defined by the OPC Data Access specification. Refer to the OPC Data Access specification for more information about these properties.

| Property Name | ID | Data Type | Description |
|---|---|---|---|
| DataType | 1 | VT_I2 | *Item Canonical Data Type*: The canonical (native) data type of the item value (property 2). |
| Value | 2 | Matches the value of property 1. | *Item Value*: The actual value of the data item. The value type matches the value of property 1. |
| Quality | 3 | VT_I2 | *Item Quality*: An indication of the reliability of the data value (property 2). |
| Timestamp | 4 | VT_DATE | *Item Timestamp*: The time the data value (property 2) was last updated. |
| AccessRights | 5 | VT_I4 | *Item Access Rights*: Indicates whether the data item can inherently be read from and/or written to. For example, inputs can generally be read from but not written to. |
| FastestScanRate | 6 | VT_R4 | *Server Scan Rate*: The best possible rate, in milliseconds, at which the server can obtain data from the data source. |
| ItemEUType | 7 | VT_I4 | *Item EU Type* |
| EUUnits | 100 | VT_BSTR | *EU Units*: A text description of the engineering units associated with the data item. |
| Description | 101 | VT_BSTR | *Item Description*: A textual description of the data item. |
| HighEU | 102 | VT_R8 | *High EU*: The highest scaled value possible for the data item. Analog data items only. |
| LowEU | 103 | VT_R8 | *Low EU*: The lowest scaled value possible for the data item. Analog data items only. |
| HighIR | 104 | VT_R8 | *High Instrument Range*: The highest possible value returned by the instrumentation. Analog data items only. |
| LowIR | 105 | VT_R8 | *Low Instrument Range*: The lowest possible value returned by the instrumentation. Analog data items only. |
| CloseLabel | 106 | VT_BSTR | *Contact Close Label*: A textual description for the data item when it is non-zero (closed). Discrete data items only. |
| OpenLabel | 107 | VT_BSTR | *Contact Open Label*: A textual description for the data item when it is in a zero (open). Discrete data items only. |
| TimeZone | 108 | VT_I4 | *Item Timezone*: The difference, in minutes, between Universal Coordinated Time (UTC) and local time. Local time + bias = UTC. |
| SoundFile | 313 | VT_BSTR | *Sound File*: A sound file associated with this data item. |

## Vendor-Defined Properties

IDs 5000 and above are custom properties defined by each server vendor.

| Property Name | ID | Data Type | Description |
|---|---|---|---|
| ItemID | 5000 | VT_BSTR | The fully qualified item name (e.g.,PressLine.Op30.GoodParts) |
| Name | 5001 | VT_BSTR | The short item name (e.g., GoodParts) |
| UsageCnt | 5002 | VT_I4 | The number of open references to this data item. |
| DataTypeStr | 5003 | VT_BSTR | The data type as a string (e.g., "VT_I4"). |
| DefDisplay | 5009 | VT_BSTR | *Default Display*: The operator display associated with this data item. |
| FgColor | 5010 | VT_I4 | *Current Foreground Color*: The foreground color in which the item should be displayed. Expressed as a COLORREF. |
| BkColor | 5011 | VT_I4 | *Current Background Color:* The background color in which the item should be displayed. Expressed as a COLORREF. |
| Blink | 5012 | VT_BOOL | *Current Blink*: Indicates whether or not the item should blink. |
| BMPFile | 5013 | VT_BSTR | *BMP File*: A graphic file associated with this data item. |
| HTMLFile | 5014 | VT_BSTR | *HTML File*: A web link associated with this data item. |
| AVIFile | 5015 | VT_BSTR | *AVI File*: A video file associated with this data item. |
| LastAccessPathNumber | 5100 | VT_I4 | One-based number of the last access path used to update the data item. |
| LastAccessPath | 5101 | VT_BSTR | A string indicating the last access path used to update the data item. |
| LastUnsolicitedFilter | 5102 | VT_BSTR | A string indicating the last unsolicited filter used to update the data item. |
| LastUnsolicitedSource | 5103 | VT_BSTR | A string indicating the last unsolicited data source used to update the data item. |
| LastUpdateSolicited | 5104 | VT_BOOL | True (non-zero) if the last value was updated as a result of a solicited request. False (zero) if it was updated by an unsolicited request. |
| Simulated | 5105 | VT_BOOL | True (non-zero) if the item's data value is simulated. |
| CurrentScanRate | 5106 | VT_UI4 | Current scan rate in milliseconds for this data item. |

# APPENDIX B: QUALITY CODES

The OPC specification requires each Data Item value to have an associated quality code. These codes fall into three main categories: good, bad and uncertain. The following sections describe all quality codes used by the Cyberlogic OPC Server. Because the low two bits in the quality code indicate the limit conditions, they are shown here as *LL*.

## Good Quality Codes

### *110000LL - Non-specific*

The value is good. There are no special conditions.

### *110110LL - Local Override*

The value has been overridden. The Cyberlogic OPC server returns this code when a simulated signal is used.

## Bad Quality Codes

### *000000LL - Non-specific*

The value is bad but the reason is unknown. This is the default quality code used by each Data Item.

### *000001LL - Configuration Error*

At runtime, the Server may detect that the physical device does not support the requested register type. For example, some devices do not support 6xxxxx registers.

### *000010LL – Not Connected*

This code normally indicates an invalid register address.

### *000011LL – Device Failure*

The Server could not complete the requested operation due to an internal failure, such as insufficient memory.

### *000101LL - Last Known Value*

Communications have failed, but the last known value is available. Note that the age of the value may be determined from the TIMESTAMP.

### *000110LL - Communication Failure*

Communications have failed and there is no last known value available.

## Uncertain Quality Codes

### *010001LL - Last Usable Value*

This code is used only with unsolicited communications. The Data Item was not updated within the Unsolicited Late Interval. The returned value should be regarded as stale.

# APPENDIX C: DATA ACCESS AUTOMATION SUPPORT

The purpose of OPC Data Access Automation is to allow applications which have an OLE Automation Interface, such as Visual Basic and Visual Basic for Applications, to access process data from OPC Data Access servers. This is done through a DLL that functions as a "wrapper", translating between the OPC interface provided by the server and the automation interface needed by the client. The wrapper does not support VB Script or Java Script, however.

## Compatibility Considerations

The DLL implementing the Data Access Automation layer was developed by the OPC Foundation and made available to OPC software vendors. Some providers made custom changes to the DLL, but did not change its name or ProgID. Consequently, all of the different versions appear to be the same, so you cannot install more than one version in a single system without causing conflicts.

The DLL provided with Cyberlogic's software, OPCDAAuto.dll, is the standard OPC Foundation file with no modifications. To avoid conflicts with other, altered versions that may be on your system, Cyberlogic's software installation program copies the file onto your hard drive but does not register it. If you wish to use our version, you must register it. Doing so will change the registration from any other version that may be on your system to the Cyberlogic version. This, in turn, may affect the operation of software that uses the other version.

## Registering the DLL

It is not necessary to copy the DLL to the System 32 directory. You can leave it in the directory where the installation program placed it and simply register it.

From the Windows Start menu, open *Command Prompt*. Change to the directory in which the DLL resides. The default installation directory is:

*C:\Program Files\Common Files\Cyberlogic Shared\OPC*

From there, type the command

*regsvr32 opcdaauto.dll*

and press <*Enter*>. The DLL will be registered and you may close the command prompt window.

# APPENDIX D: OPC XML DATA ACCESS SUPPORT

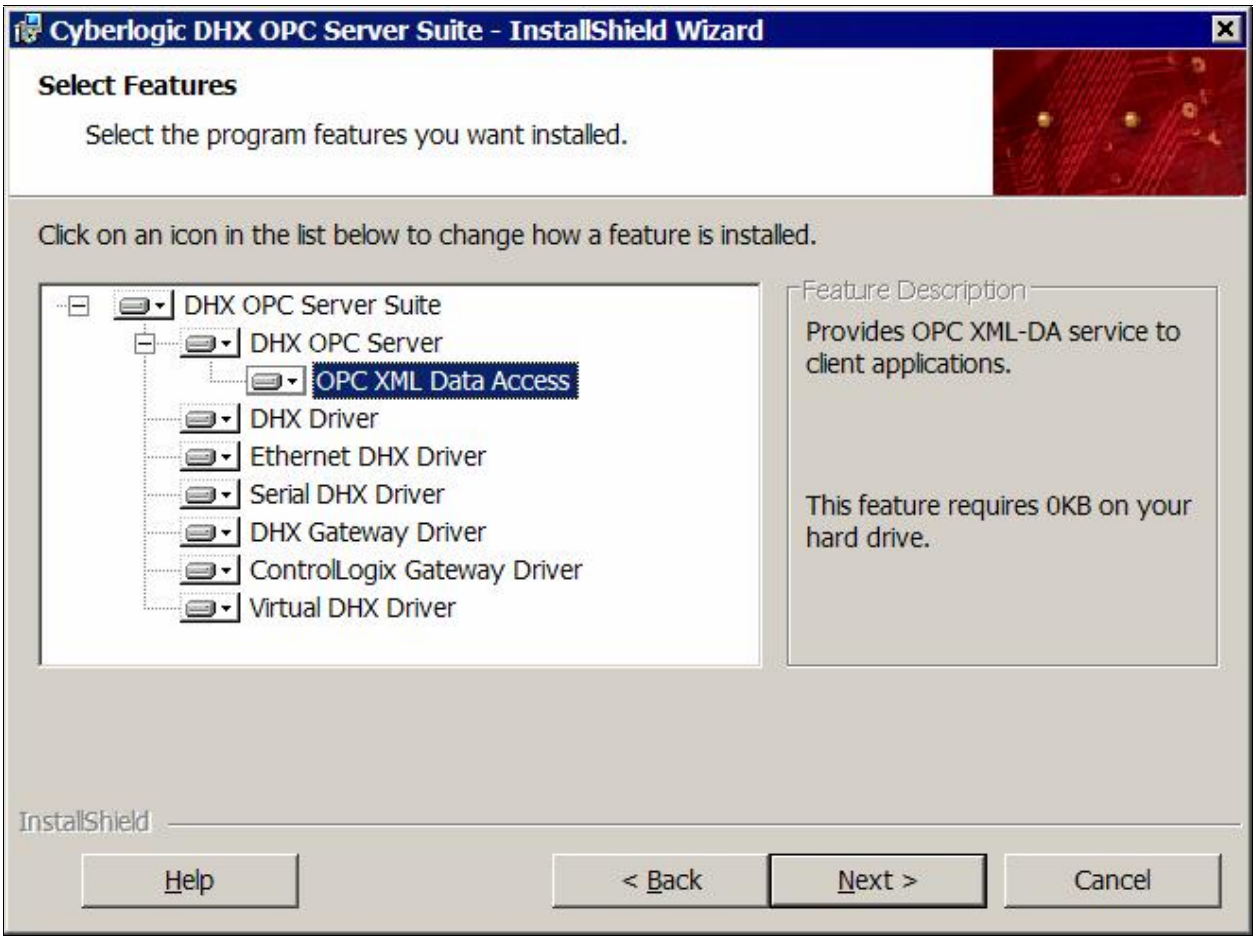
Cyberlogic's OPC Servers include support for OPC XML Data Access 1.0, but this is an optional feature that is not part of the default installation. This appendix explains how to install XML DA support and connect a client to the Server.

## Installing XML Data Access Support

To install support for XML Data Access, follow this procedure:

1. Install Windows Internet Information Services (IIS). This is a component of Windows, and therefore may require your Windows installation CD. Open the Start menu and navigate to Control Panel. Open Add or Remove Programs and finally Add/Remove Windows Components. From there you can select *Internet Information Services* and follow the prompts to complete the installation.

2. Install the Cyberlogic OPC Server XML Data Access support. To do this, run the Cyberlogic OPC Server Suite installation program and select the option to *Modify* the installation. When you are presented with the tree showing the components to install, open the DHX OPC Server or MBX OPC Server branch and select the *OPC XML Data Access* sub-branch for installation. Again, follow the prompts to complete the installation.



3. Update the DCOM security settings for the Cyberlogic OPC Server software. You must add the User account *ASPNET* to the DCOM security settings for *CybOpcRuntimeDA and CybOpcRuntimeDA*, giving ASPNET launch and access rights.

| | |
|---|---|
| **Note:** | The OPC XML Data Access layer is a client to the OPC Data Access server. It runs in the ASPNET User account. |

   To make this change, you will use the DCOM configuration editor, dcomcnfg.exe. For details on how to use this editor, refer to the *Server-Specific DCOM Configuration Issues* section of the document *OPC & DCOM: A Guide to Using the Cyberlogic OPC Server via DCOM*. A copy of this document was installed on your system along with the software and can be accessed from the Windows Start menu by navigating to the OPC Server directory and clicking on *DCOM Help*.

4. When you have finished the installation, reboot your system.

## Connecting a Client

When you open an OPC XML Data Access client application, you must specify the location of the server you want to connect to. The client may ask for an *Endpoint*, a *Server URL* or simply for a *Server*. The prompt will vary from one client to another. In any case, the form of the information you must enter is:

*http://localhost/CybOPCXML/Cyberlogic.OPCServerDA.asmx*

This is the exact form you would use if the client and server were on the same system. Otherwise, you must replace *localhost* with the IP address, computer name or web address of the server.