DHX OPC Driver Agent Help

OPC Server Driver Agent for Allen-Bradley Networks

Version 9

DHX OPC DRIVER AGENT HELP

For MicroLogix, PLC-2, PLC-3, PLC-5 and SLC Controller Families

Version 9

Document last revision date May 1, 2019

Table of ContentsIntroduction **5**

INTRODUCTION

The Cyberlogic OPC Server provides OPC Data Access, Alarms & Events and XML Data Access functions, and 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 DHX Driver Agent is one of these plug-in modules. It allows the Cyberlogic OPC Server to communicate to PLC-2, PLC-3, PLC-5, SLC-500 and MicroLogix controllers over Data Highway, Data Highway Plus, DH-485, ControlNet, Ethernet and serial DF1 connections. It is included with the DHX OPC Server Suite, DHX OPC Premier Suite and DHX OPC Enterprise Suite.

Note This document includes only the information that is specific to the DHX Driver Agent. If you need help connecting to ControlLogix controllers, refer to the ControlLogix OPC Driver Agent Help. For information on the common features of the Cyberlogic OPC Server, refer to the Cyberlogic OPC Server Help.

Compatibility and Compliance

The DHX Driver Agent is compatible with all DHX family products. The industry-standard DHX driver family, including the DHX Driver, Ethernet DHX driver, Serial DHX Driver and DHX Gateway Driver, provides the low-level communication services.

Cyberlogic OPC products provide full compliance with the OPC Foundation specifications for:

- Data Access 3.0, 2.05a and 1.0a
- Alarms & Events 1.1
- XML Data Access 1.0
- Data Access Automation 2.02

These products are tested for compliance to the OPC specifications using the latest test software from the OPC Foundation. All Cyberlogic OPC products are certified for compliance by the OPC Foundation's Independent Testing Laboratory. In addition, they are tested annually for interoperability with other OPC products at the OPC Foundation's Interoperability Workshops.

Blending DHX Supported Networks

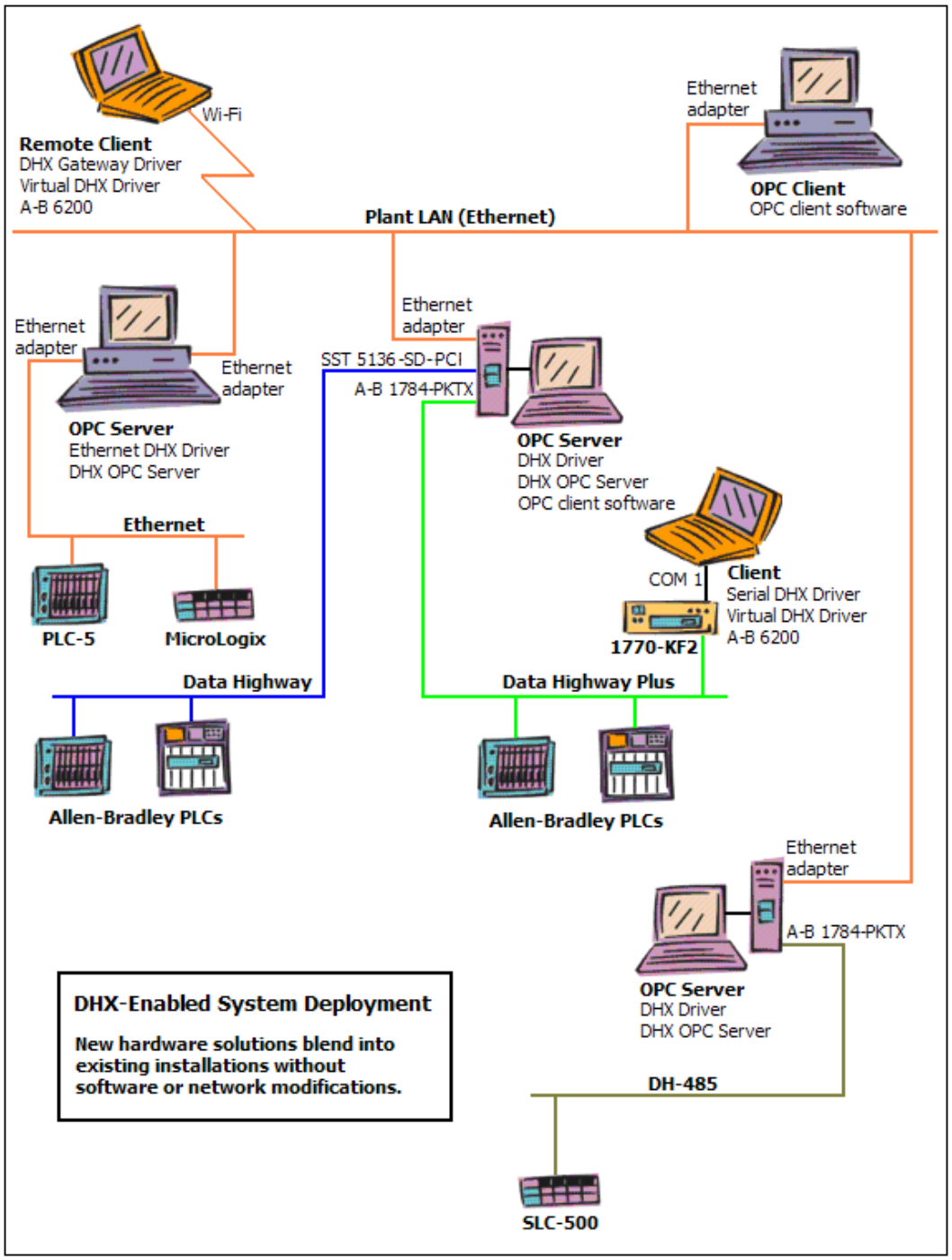
The DHX driver family provides support for all Allen-Bradley networks through a common architecture, with identical programming interfaces. This means that an application that operates with one of the DHX family drivers, such as the DHX Driver, will work with the rest of them as well. Thus, virtually all Allen-Bradley compatible software programs can

operate over all networks supported by A-B with no code modifications. You will find a complete description of the DHX family in Appendix G: DHX Architecture and Companion Products.

Migration of existing installations to new hardware products does not require the user to discard working, proven software solutions. As depicted in the diagram below, a user can mix Data Highway, Data Highway Plus, DH-485, ControlNet and Ethernet based hardware products in existing installations without losing software, network or integration investment.

The DHX family of products includes:

- DHX Driver is Cyberlogic's device driver for Data Highway, Data Highway Plus and DH-485 adapter cards from Allen-Bradley and SST.
- Ethernet DHX Driver provides Data Highway Plus emulation over TCP/IP.
- Serial DHX Driver is a full-duplex DF1 protocol driver for Data Highway, Data Highway Plus, DH-485 and ControlNet networks over serial COM port connections.
- DHX Gateway Driver works with the other DHX drivers, giving access to Data Highway, Data Highway Plus, DH-485, ControlNet and Ethernet networks from remote locations.
- ControlLogix Gateway Driver provides remote access to Allen-Bradley's Data Highway Plus network by letting you access 1756-DHRIO gateway modules in a ControlLogix chassis from a remote location.
- Virtual DHX Driver works with the other DHX drivers to permit 16-bit legacy software to run in current Windows operating systems.
- DHX OPC Server connects OPC-compliant client software applications to data sources over all Allen-Bradley networks.
- DHX SDK is a software development kit for DHXAPI, DHXAPI.Net and 6001-F1E compliant development.


Wi-Fi
Remote Client
DHX Gateway Driver
Virtual DHX Driver
A-B 6200
Ethernet adapter
OPC Client
OPC client software
Plant LAN (Ethernet)
Ethernet adapter
Ethernet adapter
OPC Server
Ethernet DHX Driver
DHX OPC Server
Ethernet adapter
SST 5136-SD-PCI
A-B 1784-PKTX
OPC Server
DHX Driver
DHX OPC Server
OPC client software
Ethernet
COM 1
Client
Serial DHX Driver
Virtual DHX Driver
A-B 6200
1770-KF2
PLC-5
MicroLogix
Data Highway
Data Highway Plus
Allen-Bradley PLCs
Allen-Bradley PLCs
Ethernet adapter
A-B 1784-PKTX
OPC Server
DHX Driver
DHX OPC Server
DH-485
SLC-500
DHX-Enabled System Deployment
New hardware solutions blend into existing installations without software or network modifications.

WHAT SHOULD I DO NEXT?

The links below will take you directly to the section of this manual that contains the information you need to configure, use and troubleshoot the DHX Driver Agent.

This document describes only the features specific to the DHX Driver Agent. For information on the common features of the Cyberlogic OPC Server, refer to the Cyberlogic OPC Server Help.

Learn How the OPC Server Works

If you are not familiar with the way that the DHX Driver Agent obtains data, you should begin by reading the Theory of Operation.

Read a Quick-Start Guide

First-time users of the DHX Driver Agent will want to refer to the Cyberlogic OPC Server Help for a quick-start guide, which walks through a typical configuration session, step-by-step.

Get Detailed Information on the Configuration Editors

Experienced users who want specific information on features of the configuration editors will find it in the Configuration Editor Reference section.

Verify That It's Working or Troubleshoot a Problem

If you have already configured the server, you should verify that it operates as expected. Refer to the Validation & Troubleshooting section for assistance. In case of communication problems, this section also provides problem-solving hints.

Print a Copy of This Document

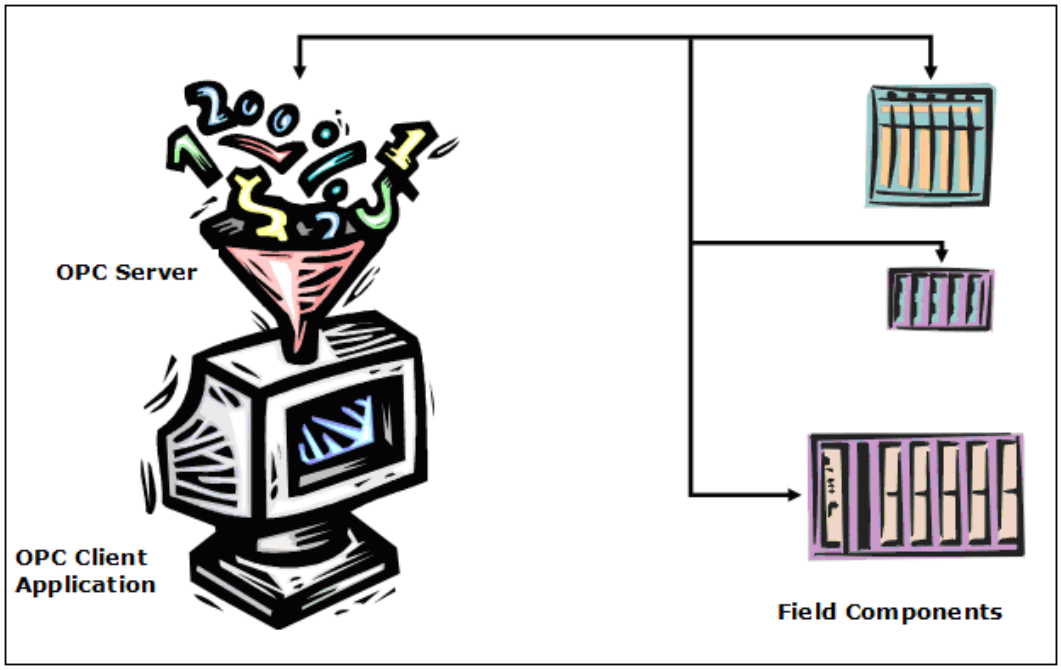
The content of this document is also provided in PDF format. PDF files can be viewed using the Adobe® Reader program, and can also be used to print the entire document.

Contact Technical Support

To obtain support information, open the Windows ***Start*** menu and go to ***Cyberlogic Suites***, and then select ***Product Information***.

THEORY OF OPERATION

This section will familiarize you with the main features of the Cyberlogic OPC Server as they relate to the DHX Driver Agent. Refer to the Cyberlogic OPC Server Help for a full discussion of the common features of the Cyberlogic OPC Server. If you are new to OPC or the Cyberlogic OPC Server, we strongly recommend that you read the OPC Tutorial first. You will find it in the Help section of your product installation.

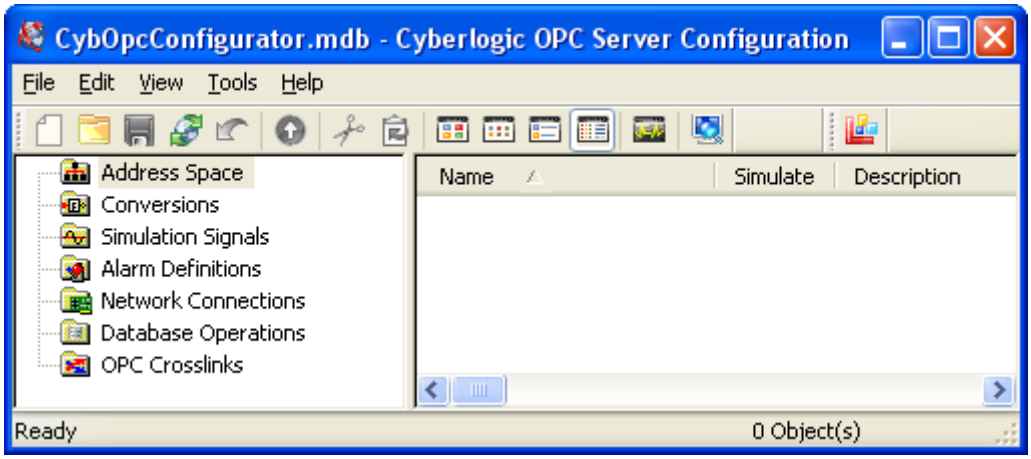


OPC servers obtain data from field components and present it, in a standard way, to OPC client applications. The basic functionality of the OPC server is the same for all types of field components and networks, but some of the communication and data-handling will vary from one family to another. The driver agent is a plug-in software module that accommodates these specific differences.

The DHX Driver Agent handles communications over all Allen-Bradley network types—Data Highway, Data Highway Plus, DH-485, ControlNet, Ethernet and serial DF1. It also provides the means to obtain and pass data from all Allen-Bradley controllers. These include PLC-2, PLC-3, PLC-5, SLC-500 and MicroLogix controllers.

The remainder of this theory section will discuss the Main Server Features you will find in the Cyberlogic OPC Server, as they relate to Allen-Bradley communications.

Main Server Features



When you open the Cyberlogic OPC Server Configuration editor, you will find several main trees. These trees represent the main areas that you will configure. Note that some are for premium features that may not be part of the product you have installed, so they will not appear in your configuration. The trees are:

- The Address Space Tree is required for most configurations. Here you will create and organize the data items that will be available to the client application, and you will define how they are updated with new information.
- The Conversions Tree is optional. In it, you can define formulas that can be used to convert raw data values obtained from the field equipment into a form that is more useful to the client. For example, you can change a transducer's voltage value into a pressure value in psi. Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.
- The Simulation Signals Tree is optional. If you want to be able to use simulated data item values instead of real values, you can create various types of simulated data functions in this tree. Simulations are often useful for troubleshooting client applications. Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.
- The Alarm Definitions Tree is another optional tree. It is used when you will interface to Alarms & Events clients. This tree allows you to define the desired alarm conditions and specify what information should be passed as they occur and clear. Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.
- The Network Connections Tree is required for all configurations. This is where you select the networks and interface devices you will use, and configure each of the field components as nodes on those networks.
- The Database Operations Tree is part of the logging feature, which is a premium feature. If this tree is in your product, you can use it to configure databases and data logging operations. Refer to the Data Logger Help for a full discussion of this tree.
- The OPC Crosslinks Tree is part of OPC Crosslink, which is a premium feature. If this tree is in your product, you can use it to configure data transfers between PLCs, between OPC servers and between PLCs and OPC servers. Refer to the OPC Crosslink Help for a full discussion of this tree.

The following sections describe these operational features of the server. Because the Network Connections Tree is normally configured first, we will start there.

Network Connections Tree

Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.

The DHX Driver Agent uses various means for connecting to their devices or networks. For a DF1 connection, a serial COM port serves that purpose. In other cases, a Data Highway Plus or Ethernet adapter card is used. The Cyberlogic OPC Server refers to all of these using the generic term "network connection". The DHX Driver Agent uses the Cyberlogic DHX family of drivers for low-level communication services to all of its network connections. Each network connection in the Cyberlogic OPC Server corresponds to a DHX device in the driver configuration, and contains all of the parameters for these devices.

The server refers to each physical device on the network as a "network node". A typical network node might be a SLC-500 controller on an Ethernet network. The server accesses the network nodes through their corresponding network connection. The network node configuration contains the communication parameters for the physical node device.

Network Automatic Configuration

The Cyberlogic OPC Server Configuration Editor can automatically detect the Allen-Bradley devices attached to the network connections and create corresponding network nodes in the server configuration file.

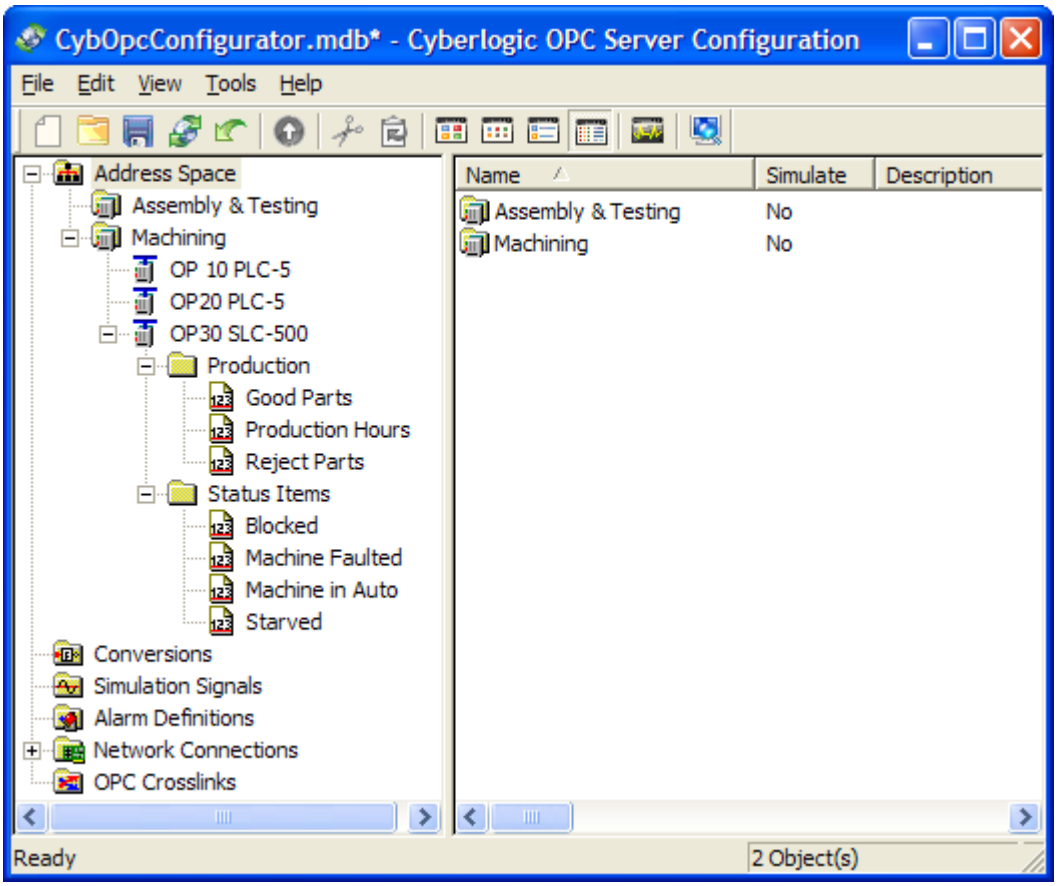
Caution!

The automatic configuration feature detects network connections and network nodes associated with the DHX devices you have configured. Before you can use automatic configuration, you must use the DHX configuration editor to create the devices that the OPC server will use.

Address Space Tree

Refer to the Cyberlogic OPC Server Help for a full discussion of this 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.

Access Paths

An access path is a logical connection to a network node. These connections, which are required for solicited communications, 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.

Each device in the server's address space can have a list of associated access paths. If there are more than one, the access path at the top of the list is the primary access path, and the rest are backups.

You can create and edit access paths on the device's Access Paths Tab.

Note

Access paths are required only for solicited communications. If you are planning to use unsolicited data updates instead, you do not have to configure any access paths. Refer to the Unsolicited Message Filters section for more information.

Unsolicited Message Filters

In addition to the more common solicited updates, the Cyberlogic OPC Server 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.

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. These filters guarantee that unsolicited messages are accepted only from trusted sources.

You can create and edit unsolicited message filters on the device's Unsolicited Message Filters Tab.

Note Unsolicited message filters are used only for unsolicited communications. If you are planning to use solicited data updates instead, you do not have to configure any unsolicited message filters. Refer to the Access Paths section for more information.

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 DHX Driver Agent supports all Allen-Bradley data types. For more information on the supported data types, refer to Appendix A: PLC Addresses. The user can individually configure each data item for solicited updates, unsolicited updates or both.

Conversions Tree

Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.

Simulation Signals Tree

Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.

Alarm Definitions Tree

Refer to the Cyberlogic OPC Server Help for a full discussion of this tree.

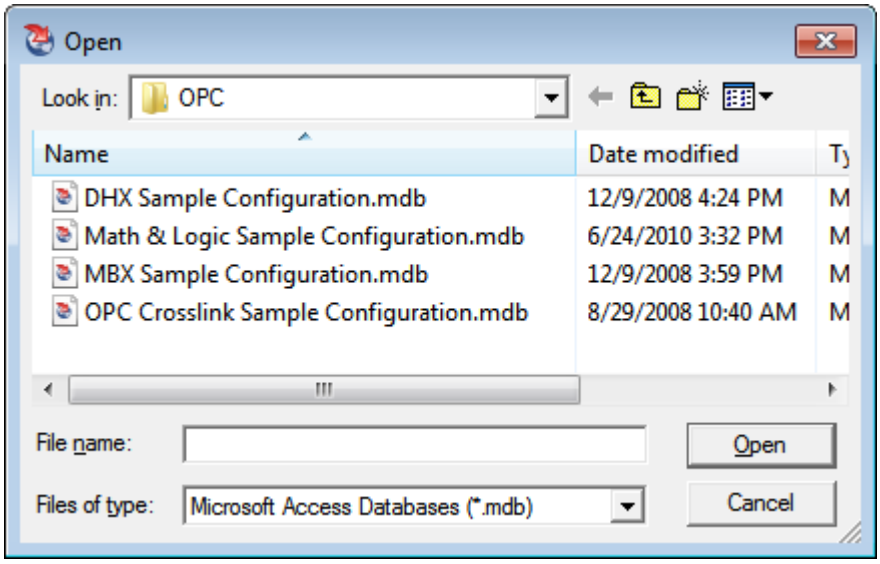
QUICK-START GUIDE

Before you can use the OPC server, you must configure it by using the OPC Server Configuration Editor. Every server requires configuration of the Network Connections tree, and most users will want to configure the Address Space tree. The remaining trees (Conversions, Simulation Signals and Alarm Definitions) are optional features used by some systems.

Sample Configuration Files

The default installation of all Cyberlogic OPC Server Suites includes a set of sample configuration files. These samples will help you to understand how to configure the OPC server for your project. In addition, the OPC Math & Logic sample provides you with numerous sample programs that you can modify and use in your system.

To open a sample configuration file from the OPC Server Configuration Editor, open the ***File*** menu and then select ***Open Sample...*** .



A browse window will open to allow you to select the configuration file you want. The available choices will depend on which OPC products you have installed.

The default location of the files is:

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

Step-By-Step Example

For a step-by-step guide through a typical configuration session, refer to the Cyberlogic OPC Server Help. After you have created the basic configuration using that procedure, the Configuration Editor Reference will explain the details of how to edit your configuration.

CONFIGURATION EDITOR REFERENCE

Before you can use the OPC server, you must configure it by using the OPC Server Configuration Editor. Every server requires configuration of the Network Connections tree, and most users will want to configure the Address Space tree. The remaining trees (Conversions, Simulation Signals and Alarm Definitions) are optional features used by some systems.

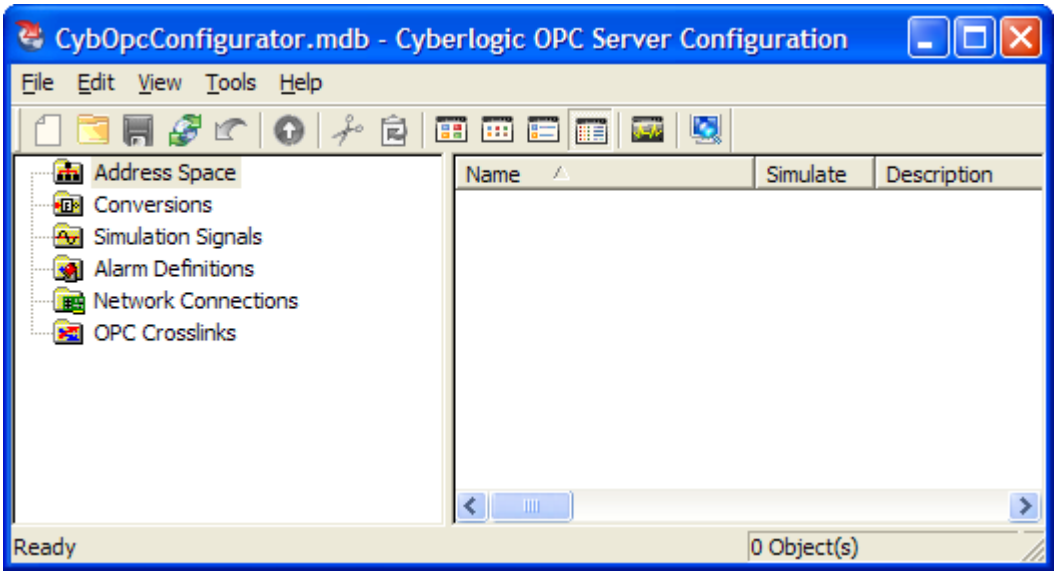
This section provides a detailed description of each of the configuration editor features. If you are a new user and want a procedure to guide you through a typical configuration session, refer to the Quick-Start Guide in the Cyberlogic OPC Server Help.

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***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

To launch the editor from the Windows ***Start*** menu, go to ***Cyberlogic Suites***, then open the ***Configuration*** sub-menu, and then select ***OPC Server***.



The left pane of the main workspace window includes the five main configuration trees. Two of these, the Network Connections and Address Space trees, are described in detail here, followed by a section on Saving and Undoing Configuration Changes.

For a discussion of the Conversions, Simulation Signals and Alarm Definitions trees and other important configuration topics including Configuration Import/Export, Editor Options and Connecting to OPC Client Software, please refer to the Cyberlogic OPC Server Help.

Network Connections

The DHX Driver Agent is a plug-in software module that permits the Cyberlogic OPC Server to communicate with Allen-Bradley networks and controllers. You will set up your OPC Server to do this by configuring appropriate network connections and network nodes in the Network Connections Tree.

The DHX Driver Agent uses the DHX drivers for its low-level communication services. A network connection in the OPC server corresponds to a DHX device in the driver architecture. A DHX device may relate to a physical network card, such as a 1784-PKTX, or an abstract object, such as an Ethernet DHX device, which behaves like a network card. A network node in the OPC server corresponds to an Allen-Bradley controller that is connected to the OPC server over one of the Allen-Bradley networks.

Creating and Deleting

There are two ways to create the network connections and network nodes: automatic and manual.

Caution!

After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

Automatic Configuration

The simplest method of configuration is automatic configuration. The automatic configuration feature detects network connections and network nodes associated with the MBX devices you have configured.

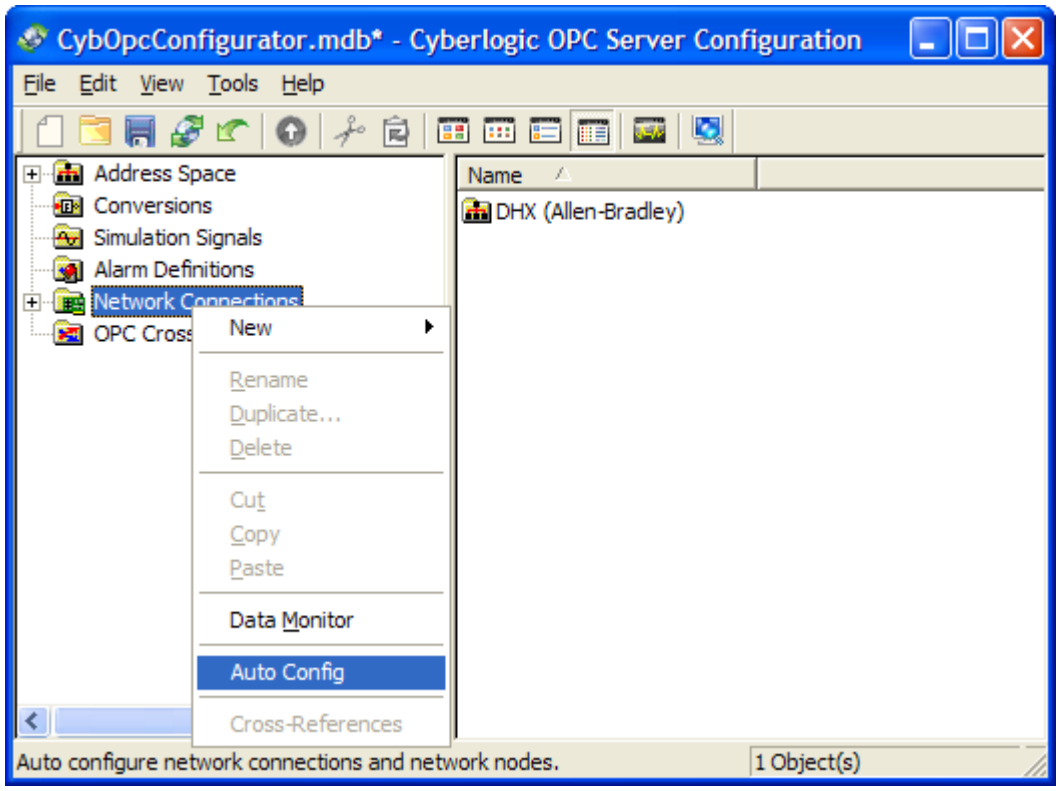
Before you can use automatic configuration, you must use the DHX configuration editor to create the devices that the OPC server will use. Refer to the driver help file for information on how to configure DHX devices.

Note

The DHX Driver Agent's automatic configuration does not identify ControlLogix controllers and reports them simply as DHX nodes of unknown type. ControlLogix nodes are normally configured using Cyberlogic's ControlLogix driver agent, which is also included as part of the DHX OPC products. However, the DHX driver agent can communicate with ControlLogix processors through the PLC-5 compatibility files. For more information refer to Appendix E: Using ControlLogix Processors.

Full Automatic Configuration

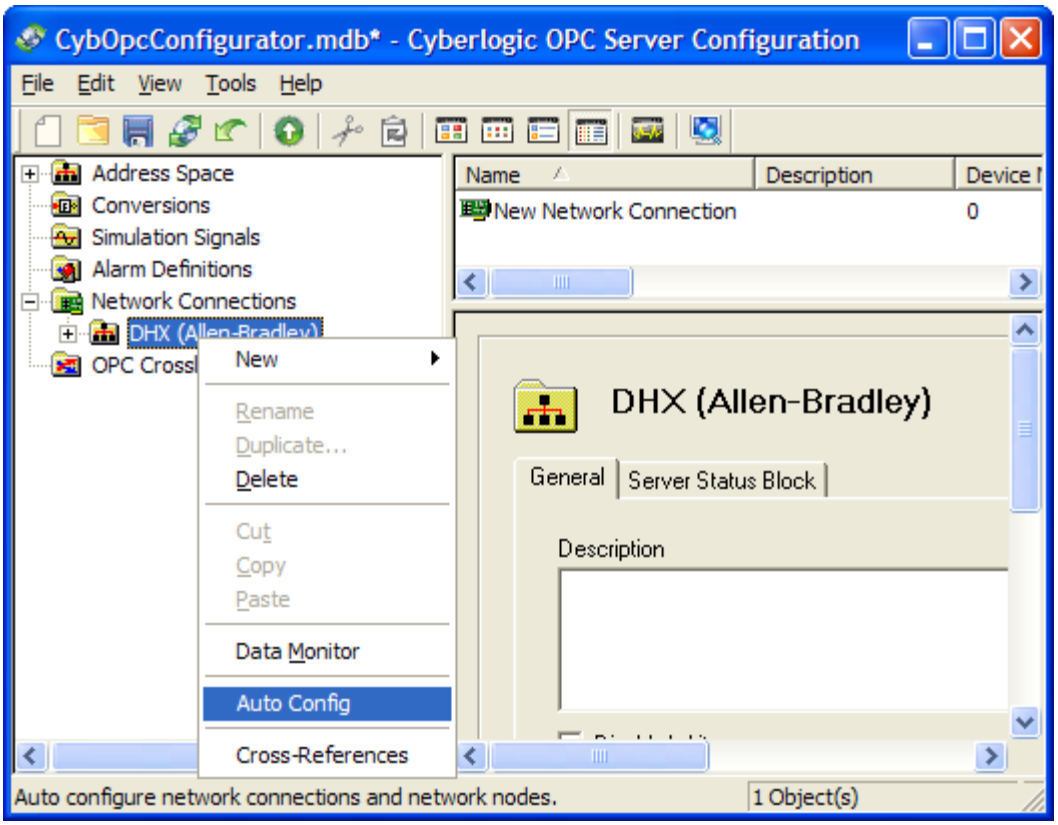
This is the most common automatic configuration procedure. It will find all network connections, and detect and configure all network nodes.



To do this, right-click on the ***Network Connections*** root folder and select ***Auto Config*** from the context menu.

Automatic Configuration of a Single Driver Agent

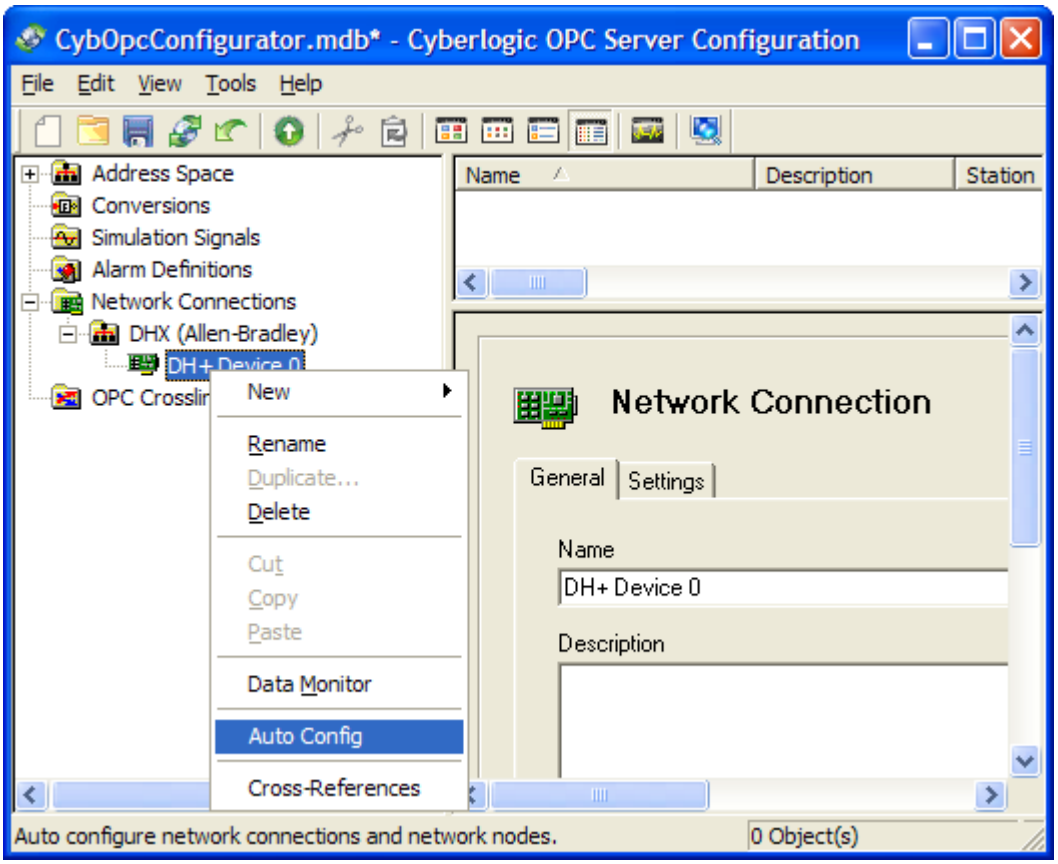
After a DHX driver agent folder has been created, you can automatically find all DHX network connections available on your system. Typically, you would do this if you did part of the configuration while not connected to the target network and want to quickly finish the configuration once you are connected.



Right-click the ***DHX (Allen-Bradley)*** folder and select ***Auto Config*** from the context menu.

Automatic Configuration of Network Nodes

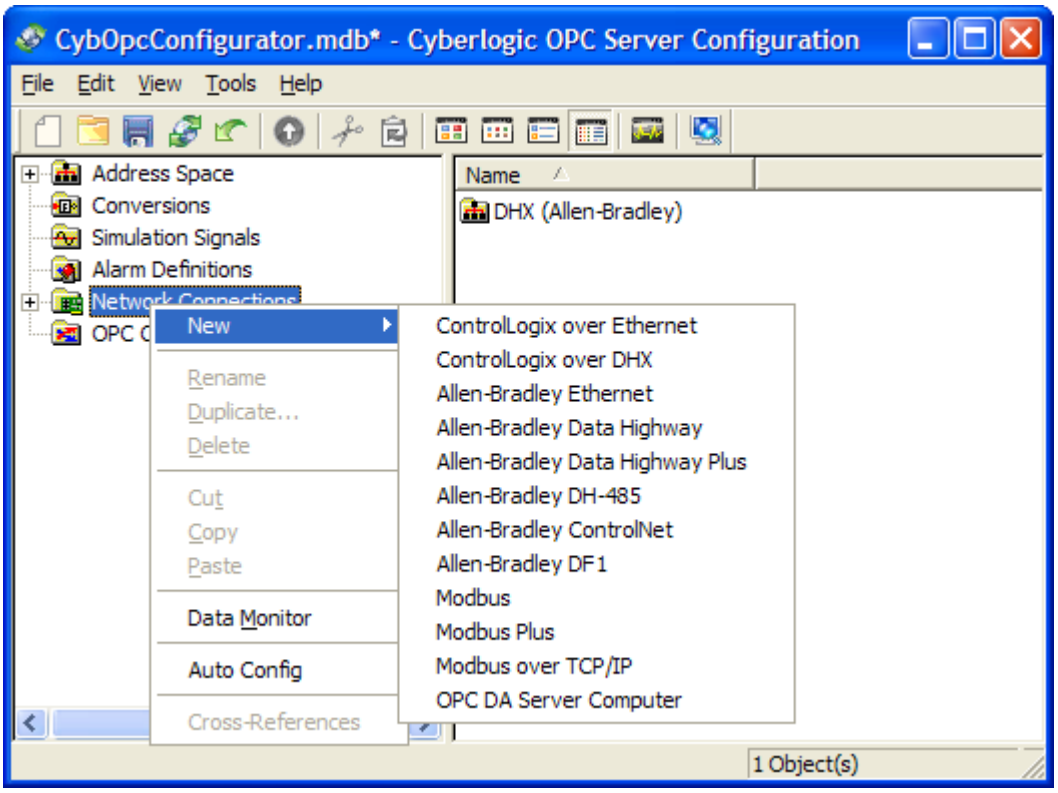
After a DHX network connection has been configured, you can automatically find and configure all of the network nodes attached to that network connection. You might do this, for example, if you add nodes to a network after it is configured and want to quickly update the configuration with the new nodes.



Right-click the specific ***DHX network connection*** and select ***Auto Config*** from the context menu.

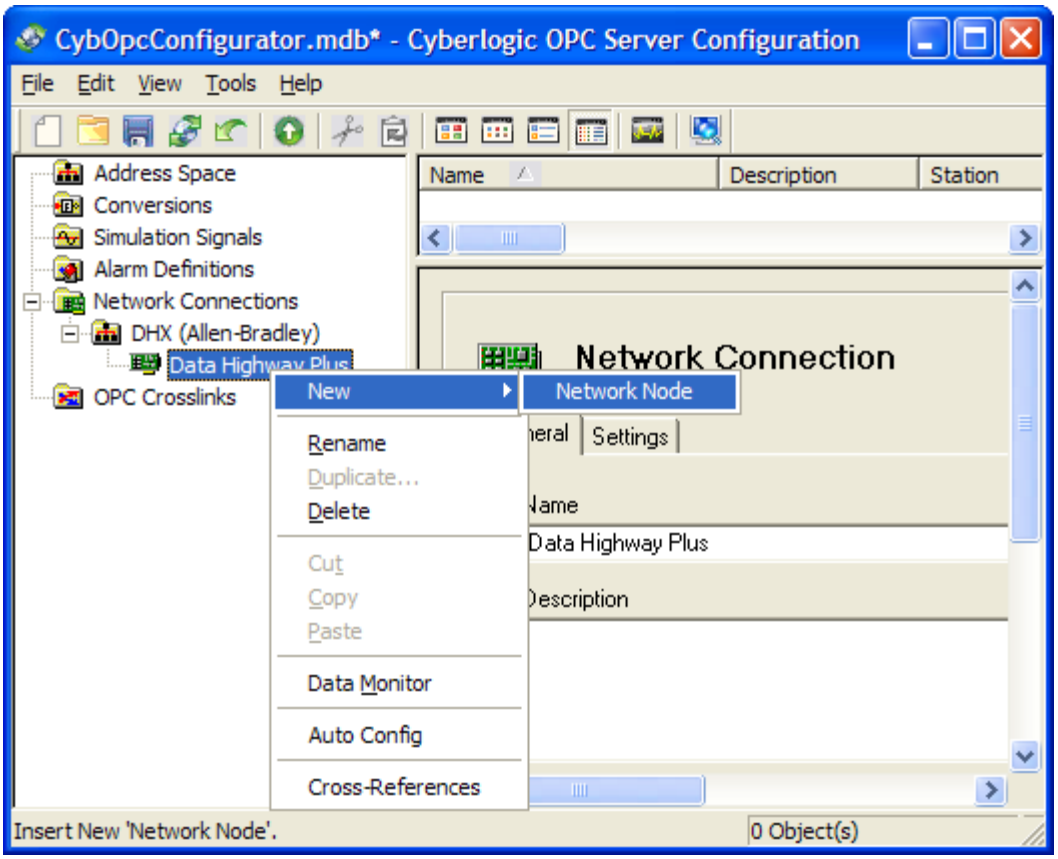
Manual Configuration

You may prefer to configure your communications 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 automatic configuration.



Right-click on the ***Network Connections*** root folder and select ***New***, then select the desired network type from the context menu.

The Editor will create the proper driver agent and network connection folders.



You can now right-click on the network connection and select ***New***, and then ***Network Node*** to manually create a network node.

Deleting

To delete the DHX Driver Agent folder, a network connection or a network node, select it and press the ***Delete key***, or right-click on it and select ***Delete*** from the context menu.

Deleting the DHX Driver Agent folder will also delete all DHX network connections and network nodes.

Deleting a DHX network connection will also delete all of its network nodes.

Note

You cannot delete a network node that is used as part of an access path for a device. If you wish to delete a network node that is in use, right-click on it and select ***Cross-References*** to obtain a list of devices that use the network node. You must then edit those devices to remove the access path that contains the network node you want to delete.

Editing the DHX Driver Agent

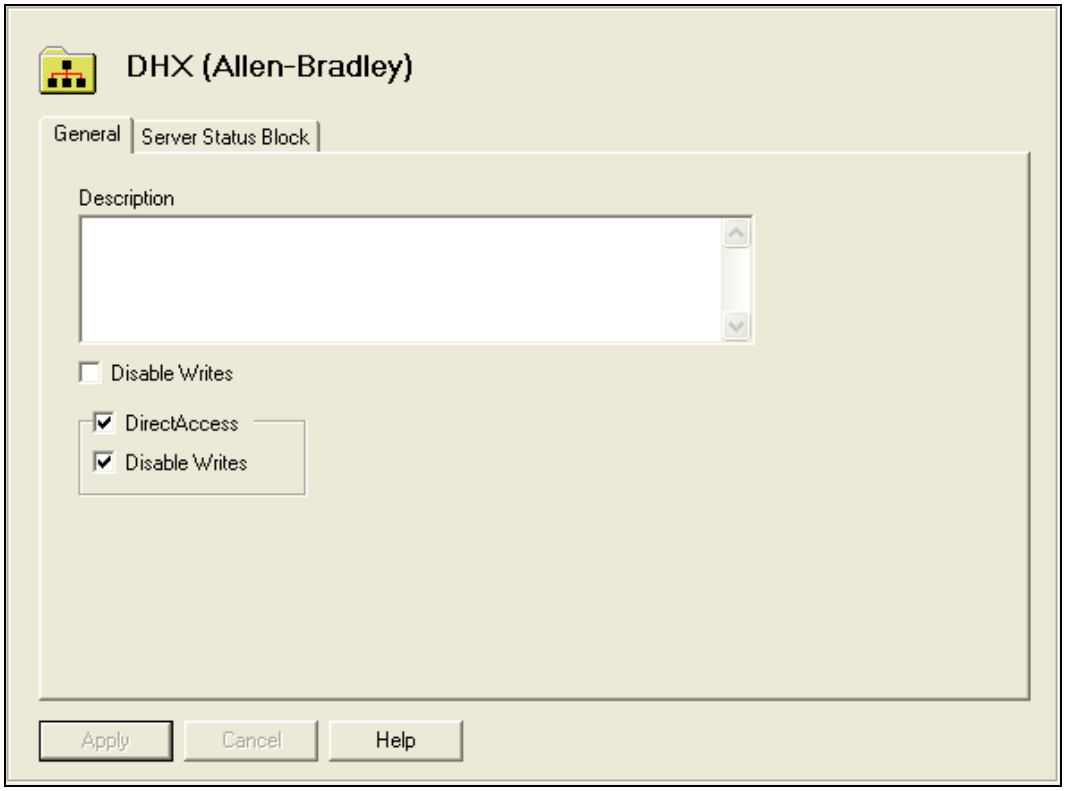
Once you have created a DHX network connection, simply click on the ***DHX (Allen-Bradley)*** folder, and the configuration screen will appear on the right side of the editor.

The DHX Driver Agent configuration has two tabs, General and Server Status Block.

Caution!

After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

General Tab



Description

This optional field can be used to describe the device. It can be up to 255 characters long.

Disable Writes

If this box is checked, the server will not write data to any of the network nodes that connect through this driver agent.

The default state is unchecked, enabling writes.

DirectAccess

If this box is checked, the user is permitted to configure DirectAccess to the network nodes that connect through this driver agent.

The default state is checked, allowing DirectAccess.

DirectAccess Disable Writes

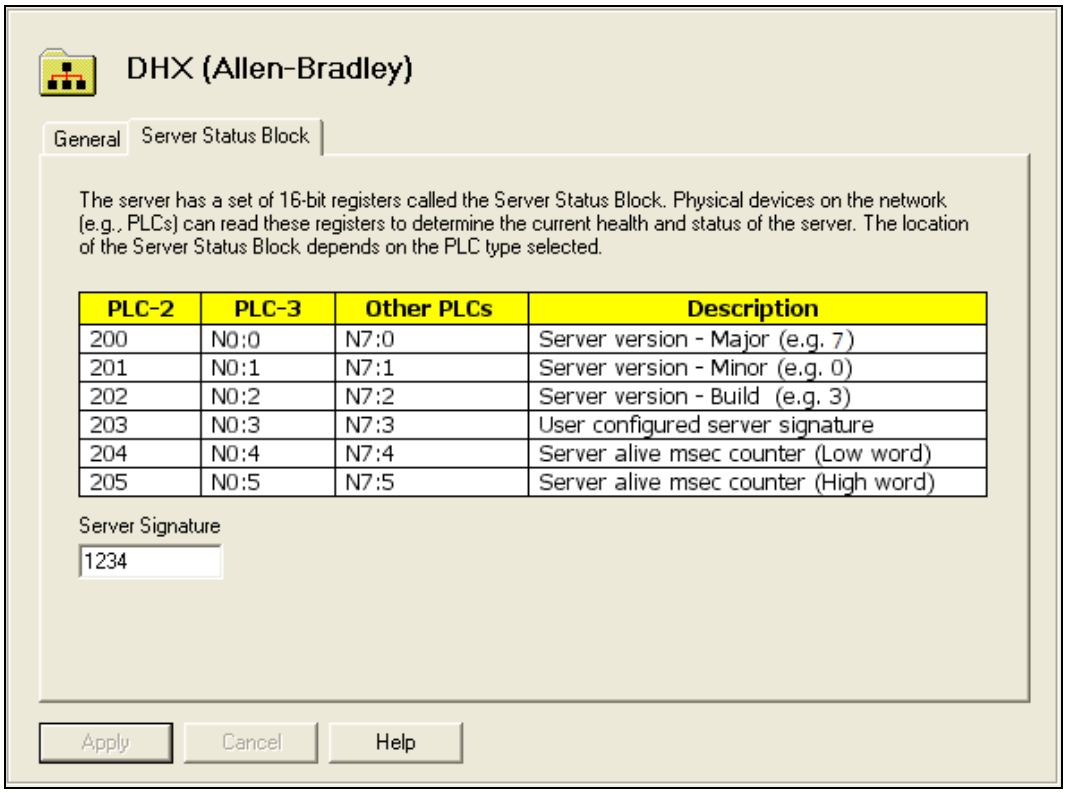
If this box is checked, the server will not write data via DirectAccess to any of the network nodes that connect through this driver agent. This does not affect writes through configured data items.

The default state is checked, disabling DirectAccess writes.

Server Status Block Tab

The Cyberlogic OPC Server has a block of 16-bit registers called the server status block. Physical devices on the network, such as PLCs, can read these registers to obtain information about the health and current status of the server. The location of the server status block depends upon the specific PLC family. The following table shows the server status block layout for the DHX driver agent:

PLC-2	PLC-3	Other PLCs	Description
200	N0:0	N7:0	Server version – Major (e.g. 7)
201	N0:1	N7:1	Server version – Minor (e.g. 0)
202	N0:2	N7:2	Server version – Build (e.g. 3)
203	N0:3	N7:3	User configured server signature
204	N0:4	N7:4	Server alive millisecond counter (Low word)
205	N0:5	N7:5	Server alive millisecond counter (High word)



Server Signature

Enter a number that will uniquely identify this server. Devices that read this signature value will then be able to identify which server they are communicating with. The default value is 1234.

Editing Network Connections

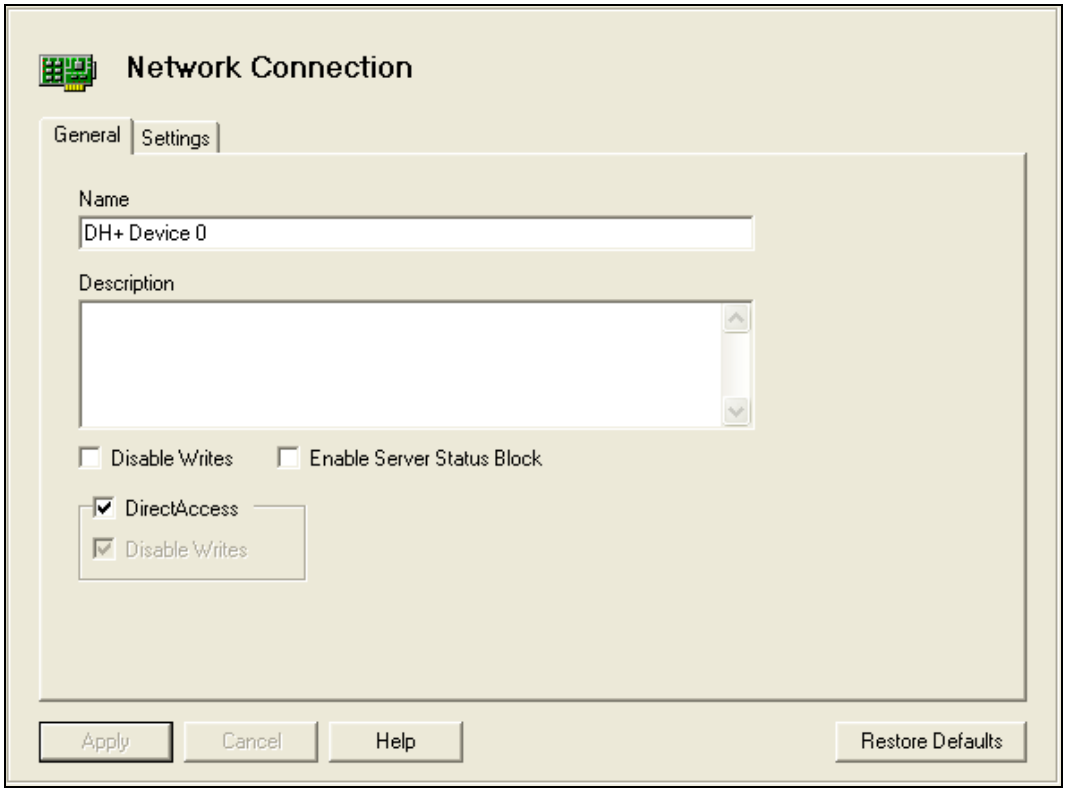
Once you have created a DHX network connection, simply select it and the configuration screen will appear on the right side of the editor.

The DHX network connection configuration has two tabs, General and Settings.

Caution!

After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, 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 this network connection. 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 network connection. It can be up to 255 characters long.

Disable Writes

If this box is checked, the server will not write data to any of the nodes on this network connection.

The default state is unchecked, enabling writes.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled at a higher level.

DirectAccess

If this box is checked, the user is permitted to configure DirectAccess to the nodes on this network connection. The default state is checked, allowing DirectAccess.

Note If the DirectAccess checkbox is grayed-out, it indicates that DirectAccess has already been disabled at a higher level.

When DirectAccess is enabled, a _Status folder will appear under this network connection in the client browser window. For more information on this folder and the status items it contains, refer to the Cyberlogic OPC Server Help.

DirectAccess Disable Writes

If this box is checked, the server will not write data via DirectAccess to any of the nodes on this network connection. This does not affect writes through configured data items. The default state is checked, disabling DirectAccess writes.

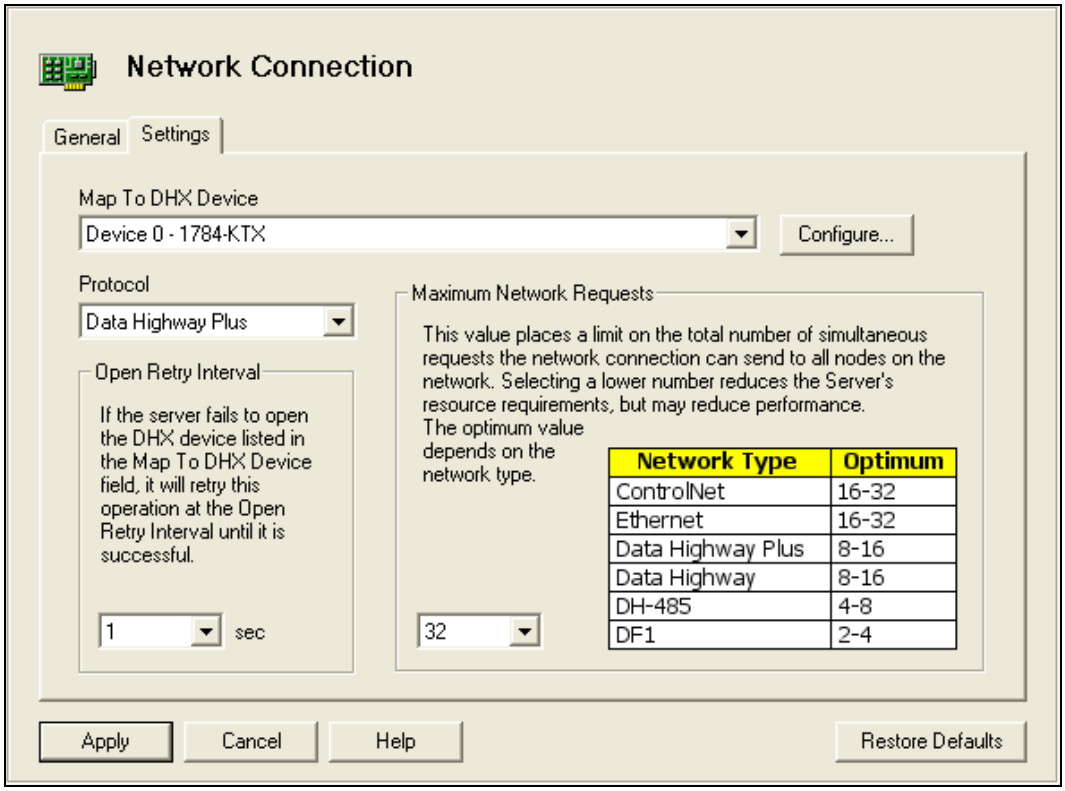
Note If the DirectAccess Disable Writes checkbox is grayed-out, it indicates that DirectAccess writes have already been disabled at a higher level.

Enable Server Status Block

If this box is checked, the network nodes under this network connection will be able to read the server status block values. For an explanation of the server status block, refer to the Server Status Block Tab section.

The default state is unchecked, disabling access to the server status block.

Settings Tab



Map To DHX Device

Click the drop-down button to select the DHX device this network connection will use to communicate.

The data field will tell you if the DHX device is not configured or if it is configured for a different protocol than you have selected for the network connection.

Configure…

Click this button to run the DHX Driver Configuration Editor, allowing you to modify the configuration of the DHX devices and to create new ones. For more information on configuring DHX devices, refer to the help files for each of the DHX family drivers.

Protocol

Click the drop-down button to select the specific protocol used by this network.

Open Retry Interval

The server will try to open the DHX device shown in the Map To DHX Device field. If it fails, it will retry repeatedly until it succeeds. This field specifies the interval at which the server will attempt these retries.

The valid range for the Open Retry Interval is 1-60 seconds, and the default is 1 second.

Maximum Network Requests

This value defines the maximum number of simultaneous transactions that the server will allow for the nodes on this network connection. The number of concurrent requests may also be limited for each network node on its Optimizations Tab.

The optimum value to use depends upon the network type. The slowest network in the path to the network node should dictate the range to use. For example, if the OPC server has a 1GB Ethernet connection to a bridge, and the bridge has a 9600 baud serial connection to the network node, the DF1 range should be used, not the Ethernet range.

Use the following table as a guideline when selecting this value.

Network Type	Optimum Range
ControlNet	16-32
Ethernet	16-32
Data Highway Plus	8-16
Data Highway	8-16
DH-485	4-8
DF1	2-4

Caution!

Lower values reduce the server's resource requirements, but may reduce performance.

Selecting values above the recommended range consumes more system resources, but typically does not improve the performance of the server, and may actually harm performance.

Refer to Appendix F: Configuring Maximum Concurrent Requests for more information and examples.

Editing Network Nodes

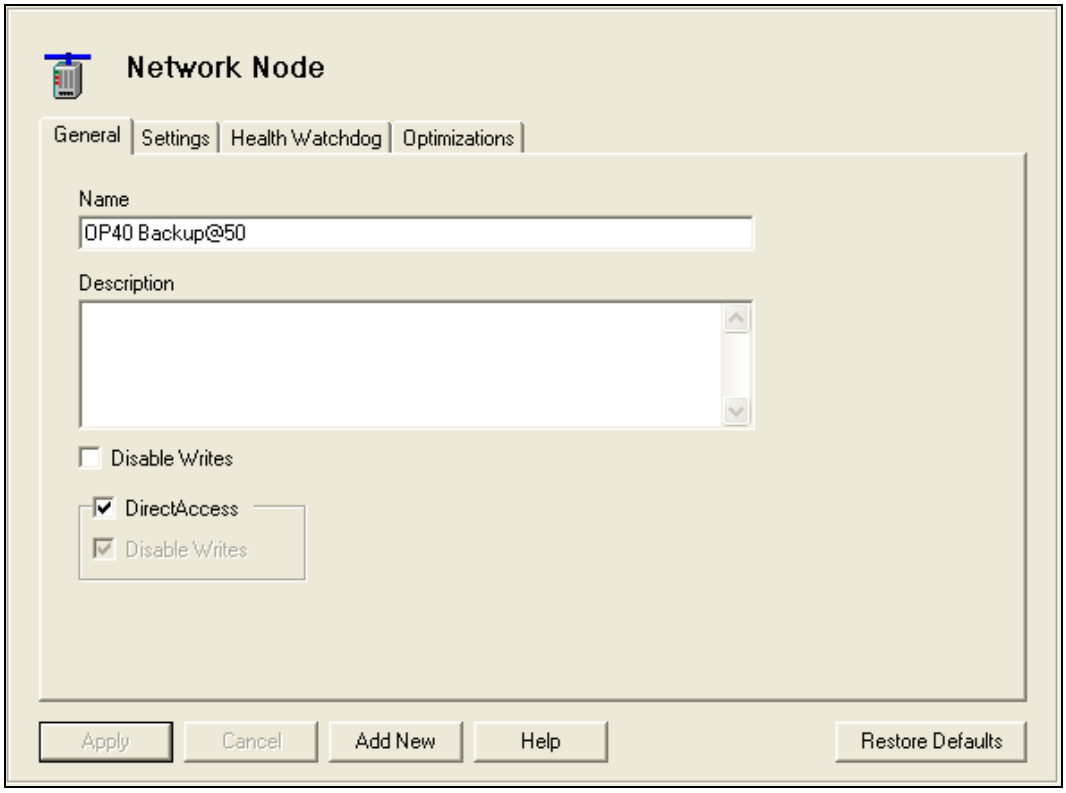
Once you have created a DHX network node, simply select it and the configuration screen will appear on the right side of the editor.

The DHX network node configuration has four tabs, General, Settings, Health Watchdog and Optimizations.

Caution!

After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, 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 network node. It can be up to 50 characters long, may contain spaces, but must not begin with a space. It also may not contain any periods.

Description

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

Disable Writes

If this box is checked, the server will not write data to this node. The default state is unchecked, enabling writes.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled at a higher level.

DirectAccess

If this box is checked, the user is permitted to configure DirectAccess to this node. The default state is checked, allowing DirectAccess.

Note If the DirectAccess checkbox is grayed-out, it indicates that DirectAccess has already been disabled at a higher level.

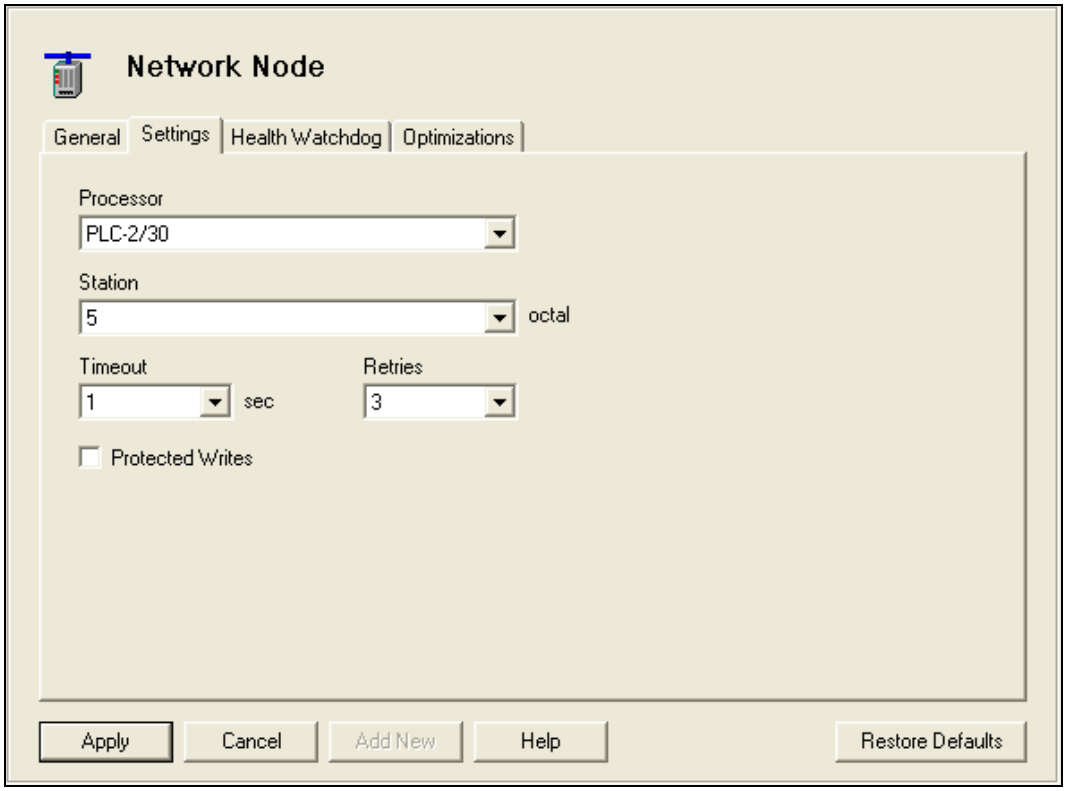
When DirectAccess is enabled, a _Status folder will appear under this network node in the client browser window. For more information on this folder and the status items it contains, refer to the Cyberlogic OPC Server Help.

DirectAccess Disable Writes

If this box is checked, the server will not write data via DirectAccess to this node. This does not affect writes through configured data items. The default state is checked, disabling DirectAccess writes.

Note If the DirectAccess Disable Writes checkbox is grayed-out, it indicates that DirectAccess writes have already been disabled at a higher level.

Settings Tab



Processor

This field identifies the physical device type of this network node. You may select a specific PLC model or a PLC family. Choose a family if you are unsure of the exact model in use. The server uses this information to optimize network communications.

Station

This is the node's network address. The interpretation of this address varies according to the type of network. For more information on node addressing, see the help files for the DHX family drivers.

Timeout

This is the amount of time that the server will wait to receive a reply to a command message. If the server does not receive a reply within that interval, it cancels the transaction and marks it as timed out. This interval is specified in seconds.

Retries

This is the number of times the server will reattempt each transaction that fails. The health watchdog uses this value to determine when to mark the node as unhealthy. Refer to the Health Watchdog Tab section for more information on this feature.

Protected Writes

This checkbox applies only to PLC-2 family controllers and will not be present for other types.

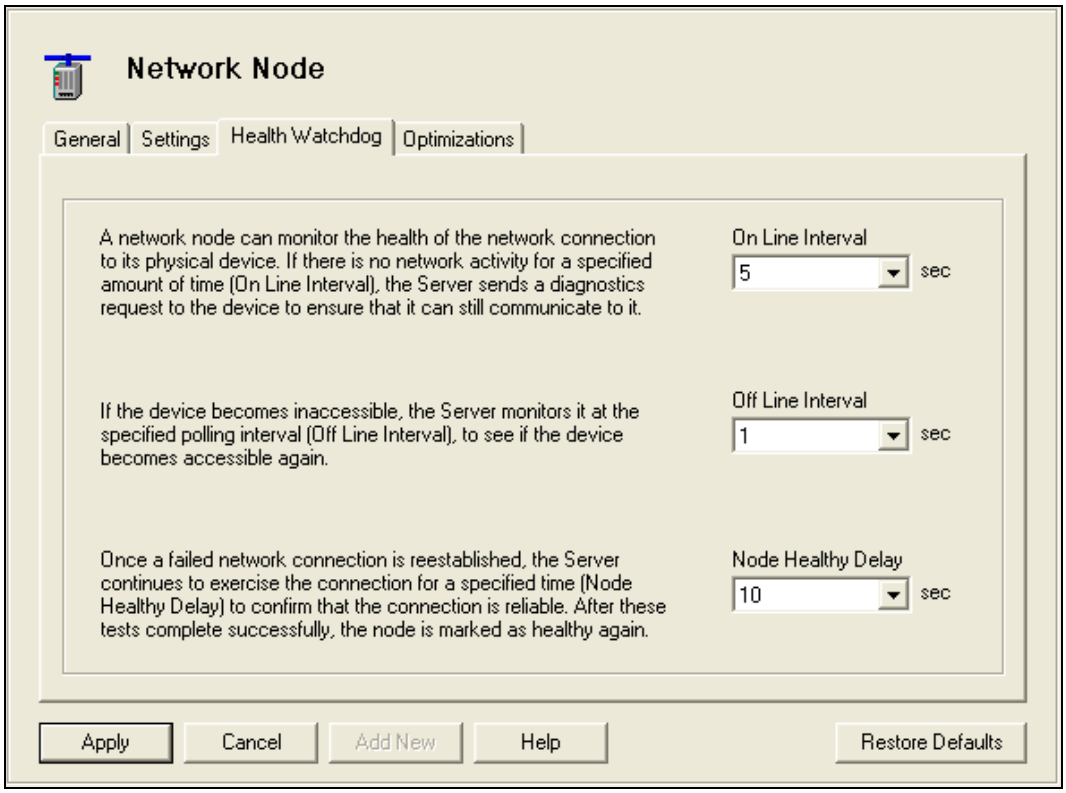
When this box is checked, the server will use the Protected Write command for all write operations to the controller. This command writes words of data into limited areas of the PLC data table memory. Its access is limited by memory access rungs in the communication zone of the PLC ladder diagram program.

When the box is not checked, the server uses Unprotected Write commands.

Health Watchdog Tab

Each network node monitors the health of the connection to its physical device. If there is no communication for a long time, the server sends a diagnostic request to the device to see if it can still communicate. If it cannot, the server will re-check the connection until communication is reestablished. Once a failed network connection is reestablished, the server continues to exercise the connection until it is satisfied that the connection is reliable. After this, the node is marked as healthy again.

The Health Watchdog tab allows you to configure the time intervals associated with these tests.



On Line Interval

You may select a value in the range of 1-60 seconds or None.

If there is no traffic to a healthy node for the specified length of time, the server will send a status request to the node to verify that it is still online. Selecting *None* disables the on-line health monitoring.

If you are using a slow network, such as DF1, and you are not using redundant access paths, you might want to disable health monitoring to reduce some of the network traffic. In that case, the OPC server will always report the status of the node as healthy.

Off Line Interval

You may select a value in the range of 1-60 seconds.

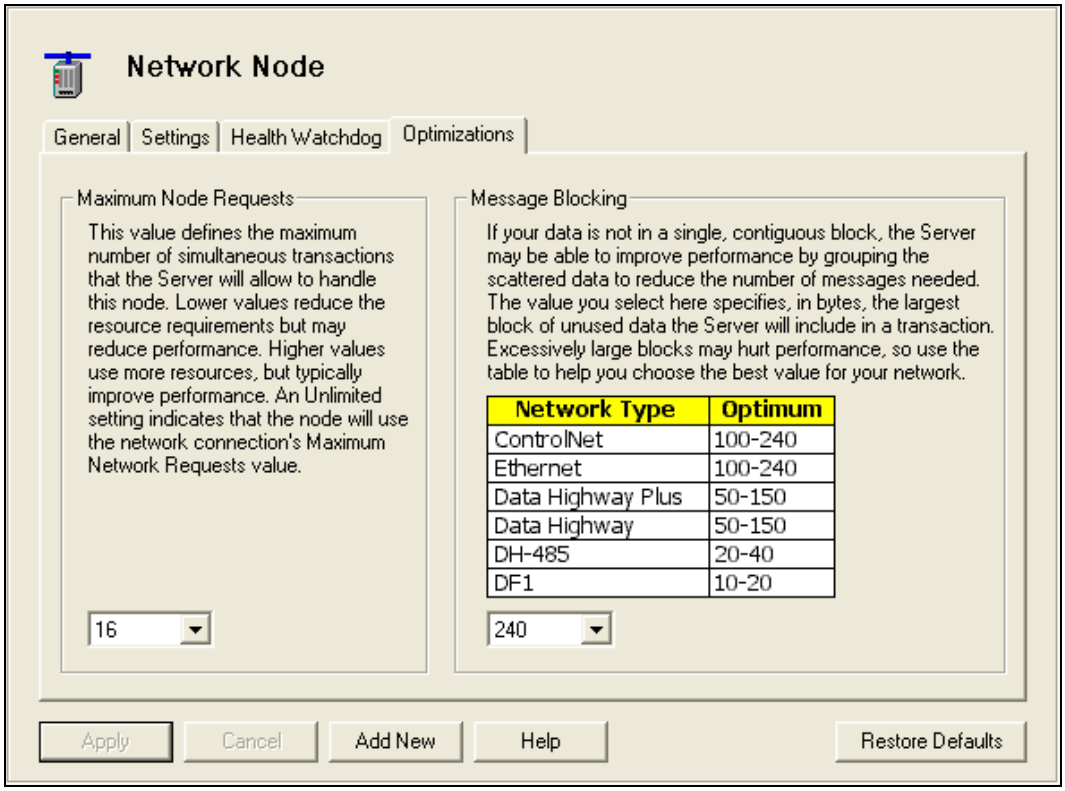
Once communication to a node has failed, the server will send status requests to the node at this interval to determine whether it can communicate.

Node Healthy Delay

You may select a value in the range of 1-60 seconds.

Once the server reestablishes communication to an unhealthy node, it waits for the communication to stay active for this length of time before considering the node to be healthy. The server then returns the node to service.

Optimizations Tab



Maximum Node Requests

This value defines the maximum number of simultaneous transactions that the server will allow for this node. The number of concurrent requests is also limited for the network connection on its Settings Tab.

Lower values reduce the resource requirements but may reduce performance. Higher values use more resources, but typically improve performance. However, an excessively high setting may reduce performance. Setting this selection to Unlimited causes the node to use the network connection's Maximum Network Requests value.

Note

Setting this value to a lower number may prevent overloading the PLC with messages and allow better operation of other applications, such as PLC programming software, that access the same PLC.

Refer to Appendix F: Configuring Maximum Concurrent Requests for more information and examples.

Message Blocking

Typically, the data you will want to access will not be in a single, contiguous block. For the server to access only the information you really require, it may need to make many small transactions. In that case, the server may obtain better performance by grouping

the data into fewer, larger blocks. Although this will mean that the system will transfer unneeded information, the overall throughput may be improved because of the reduced overhead.

This field allows you to tell the server how to group the scattered data. The value you enter specifies, in bytes, the largest block of extra data that the server will include in a transaction. The optimum value to use will depend upon the type of network. Faster networks can handle larger gaps; slower networks will get better performance with smaller gaps.

The slowest network in the path to the network node should dictate the Message Blocking value to use. For example, if the OPC server has a 1GB Ethernet connection to a bridge, and the bridge has a 9600 baud serial connection to the network node, the DF1 range should be used.

Network Type	Optimum Range
ControlNet	100-240
Ethernet	100-240
Data Highway Plus	50-150
Data Highway	50-150
DH-485	20-40
DF1	10-20

Address Space

The Address Space Tree describes the hierarchical address structure of the Cyberlogic OPC Server. The branches of the tree are Device Folders, Devices and Folders. Its "leaves" are Data Items. The intent of this structure is to permit the user to organize the data items into logical groups.

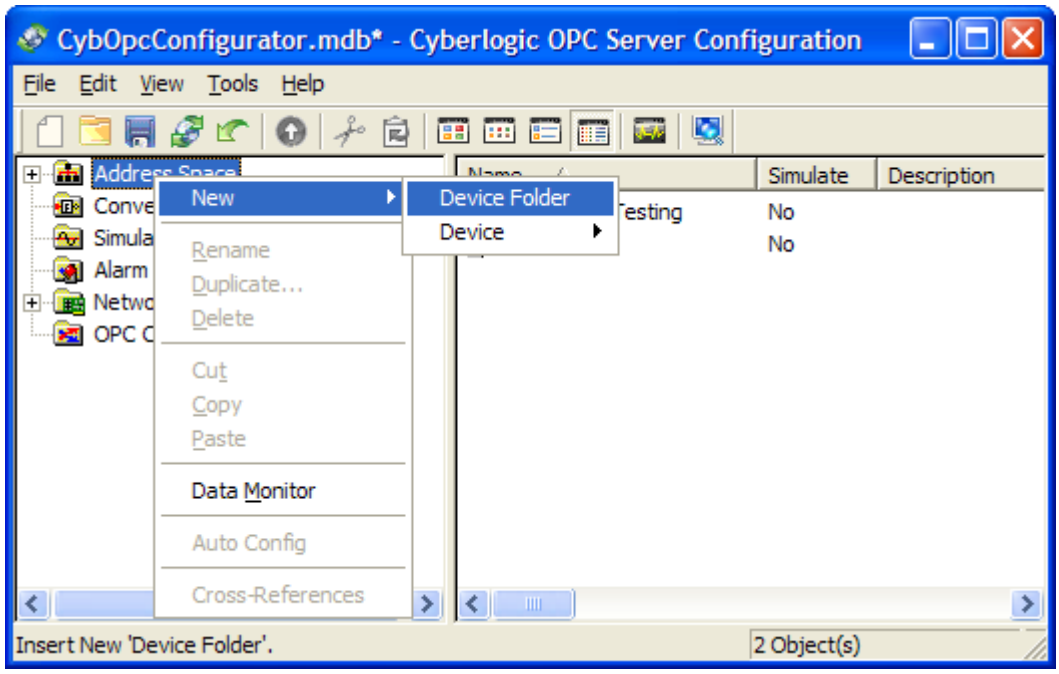
Device Folders

A device folder groups devices and other device folders. You can place a device folder directly under the Address Space root folder or under another device folder, up to four levels deep.

Caution!

After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

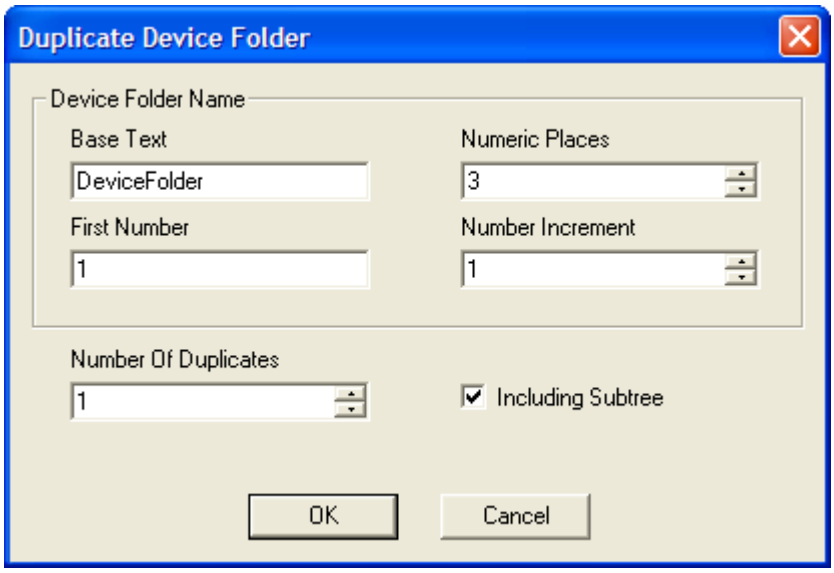
Creating a New Device Folder



Right-click on the Address Space root folder or an existing device folder. Select ***New*** and then ***Device Folder*** from the context menu.

Duplicating a Device Folder

To speed up the creation of similarly-configured device folders, you can create multiple device folders in a single operation by duplicating an existing one. To do this, right-click on an existing device folder and select ***Duplicate...*** from the context menu.



The above dialog box opens. You must specify how the duplicates are to be named by entering values for the ***Base Text***, ***First Number***, ***Numeric Places*** and ***Number Increment*** fields. To generate names for the duplicated device folders, the editor

begins with the base text and appends a number to it. The first duplicate uses the selected First Number value with the specified number of digits. This number is then incremented by the specified number for each of the remaining duplicates.

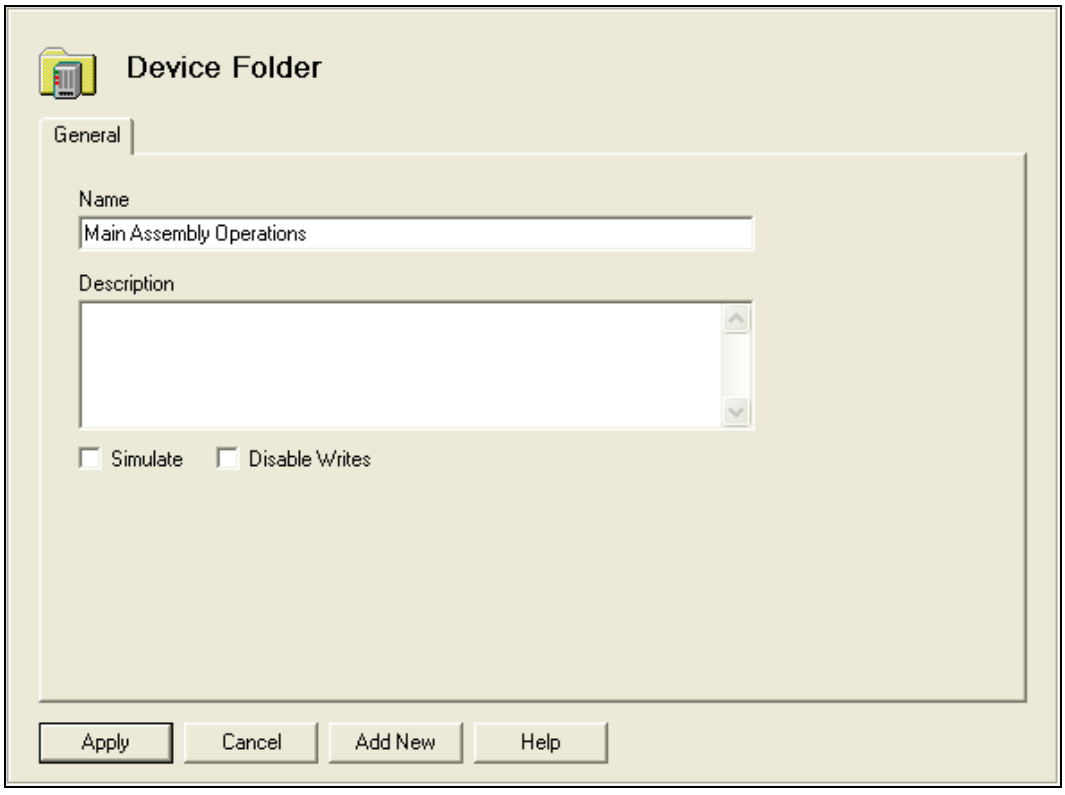
As an example, if Numeric Places is 3 and First Number is 2, the number 002 will be appended to the base text.

Use the ***Number Of Duplicates*** field to specify the number of device folders you wish to create. If you want to duplicate all branches within the original device folder, check the ***Including Subtree*** checkbox.

Deleting a Device Folder

To delete an existing device folder, select it and press the ***Delete key***, or right-click on the device folder and select ***Delete*** from the context menu.

General Tab



Name

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

Description

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

Simulate

Checking this box enables data simulation for all data items found at this level or below. This provides a quick way to switch between real and simulated data for a large number of data items. Refer to the Cyberlogic OPC Server Help for a full discussion about simulating data.

Note If the Simulate checkbox is grayed-out, it indicates that simulation has already been selected at a higher level.

Disable Writes

Checking this box disables write requests for all data items found at this level or below. By default, this box is not checked and writes are enabled.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled a higher level.

Devices

A device in the address space represents a source of data to which the server communicates. This data source may be a PLC, a group of PLCs or other physical data sources. You can place devices 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.

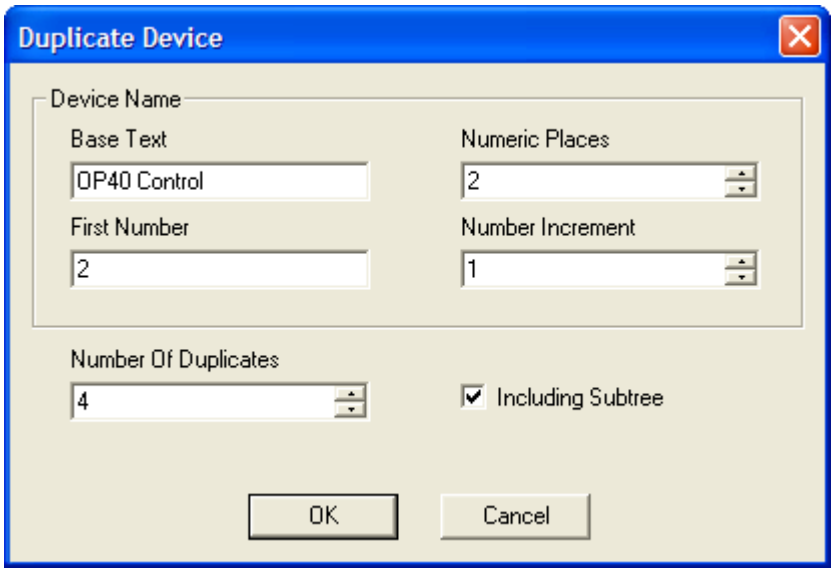
Caution! After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

Creating a New Device

Right-click on the Address Space root folder or an existing device folder. Select ***New*** and then ***Device*** from the context menu.

Duplicating a Device

To speed up the creation of similarly-configured devices, you can create multiple devices in a single operation by duplicating an existing one. To do this, right-click on an existing device and select ***Duplicate...*** from the context menu.



The above dialog box opens. You must specify how the duplicates are to be named by entering values for the ***Base Text***, ***First Number***, ***Numeric Places*** and ***Number Increment*** fields. To generate names for the duplicated devices, the editor begins with the base text and appends a number to it. The first duplicate uses the selected First Number value with the specified number of digits. This number is then incremented by the specified number for each of the remaining duplicates.

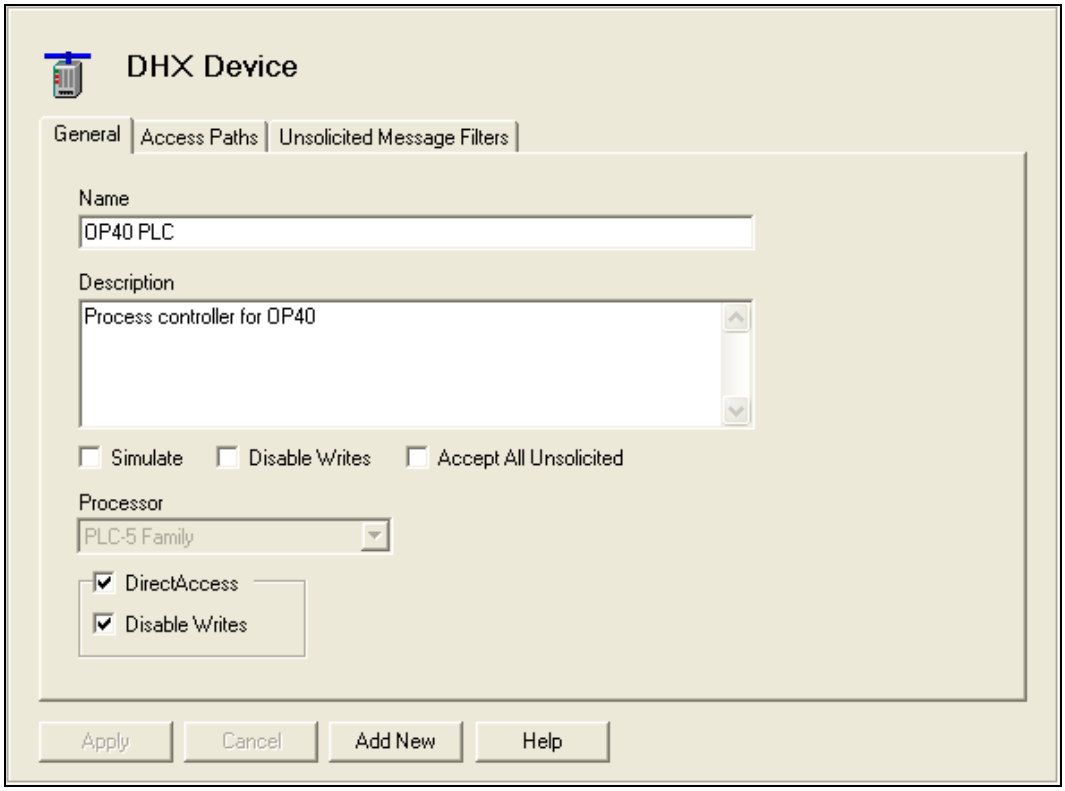
As an example, if Numeric Places is 3 and First Number is 2, the number 002 will be appended to the base text.

Use the ***Number Of Duplicates*** field to specify the number of devices you wish to create. If you want to duplicate all branches within the original device, check the ***Including Subtree*** checkbox.

Deleting a Device

To delete an existing device, select it and press the ***Delete key***, or right-click on the device and select ***Delete*** from the context menu.

General Tab



Name

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

Description

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

Simulate

Checking this box enables data simulation for all data items found at this level or below. This provides a quick way to switch between real and simulated data for a large number of data items. Refer to the Cyberlogic OPC Server Help for a full discussion about simulating data.

Note If the Simulate checkbox is grayed-out, it indicates that simulation has already been selected at a higher level.

Disable Writes

Checking this box disables write requests for all data items found at this level or below. By default, this box is not checked and writes are enabled.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled a higher level.

Accept All Unsolicited

When this box is checked, the server will ignore the unsolicited message filters and will accept all unsolicited messages. By default, this box is not checked and unsolicited messages will be required to pass the filter criteria.

For more information on unsolicited message filters, refer to the Unsolicited Message Filters Tab section.

Processor

Select the processor family from this drop-down box. Although you will be able to configure access paths to different PLCs within the same device, all of these PLCs must be of the same family. This is necessary because of the different addressing techniques used by different families.

Note After you make the initial edits and click the Apply button, this selection will no longer be available for you to change.

DirectAccess

If this box is checked, the user is permitted to configure DirectAccess to the nodes associated with this device. The default state is checked, allowing DirectAccess.

When DirectAccess is enabled, a _Status folder will appear under this device in the client browser window. For more information on this folder and the status items it contains, refer to the Cyberlogic OPC Server Help.

DirectAccess Disable Writes

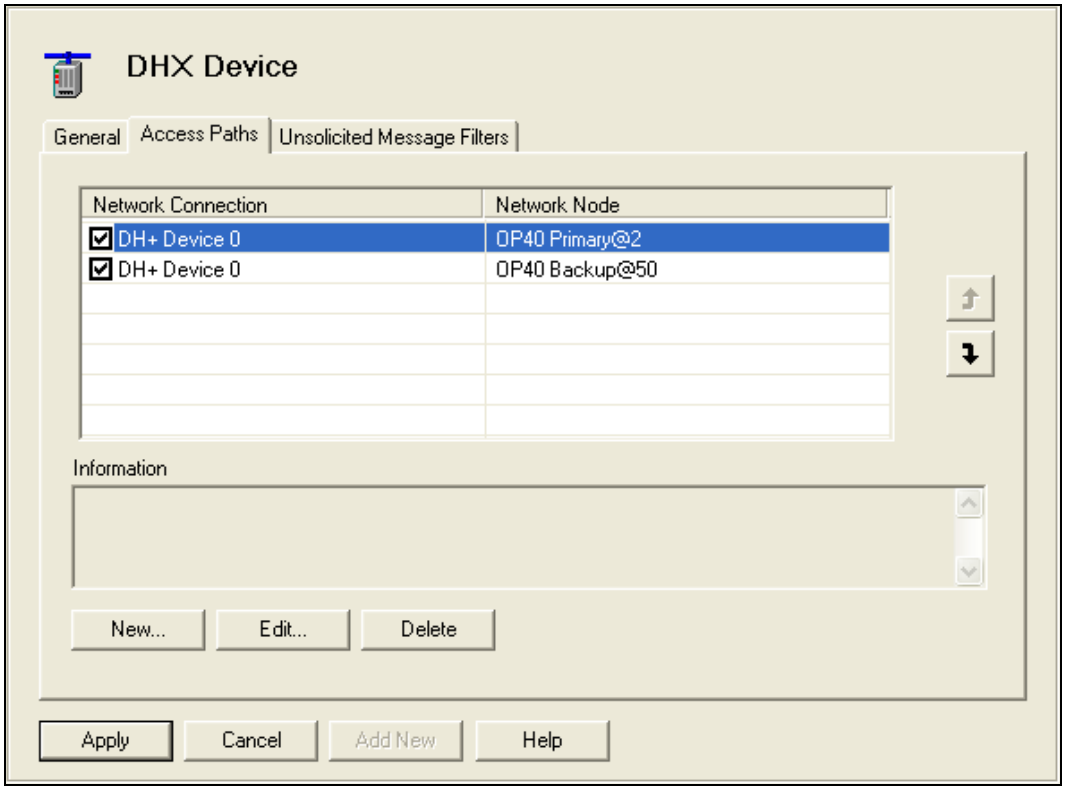
If this box is checked, the server will not write data via DirectAccess to any of the nodes associated with this device. This does not affect writes through configured data items. The default state is checked, disabling DirectAccess writes.

Access Paths Tab

Each device has an associated list of access paths. Depending upon your network configuration, the access paths may include multiple paths to the same PLC, paths to different PLCs, or some combination of the two. The access path at the top of the list is

the primary access path; the rest are backups. If the current access path fails or is disabled, the server switches to the highest access path that is available and enabled.

Note Access paths are required for solicited communications only. If you are planning to use only unsolicited data updates, no access paths need to be configured.



Enable Checkbox

To the left of each access path is a checkbox that, when checked, enables the access path. The server uses only enabled access paths.

Network Connection

The Network Connection column displays the network connection associated with each of the access paths.

Network Node

The Network Node column displays the network node associated with each of the access paths.

Up and Down Arrows

The server assigns priority to the access paths from top to bottom, with the access path at the top having the highest priority. Use the up and down arrows on the right side of the list box to adjust the priorities as needed.

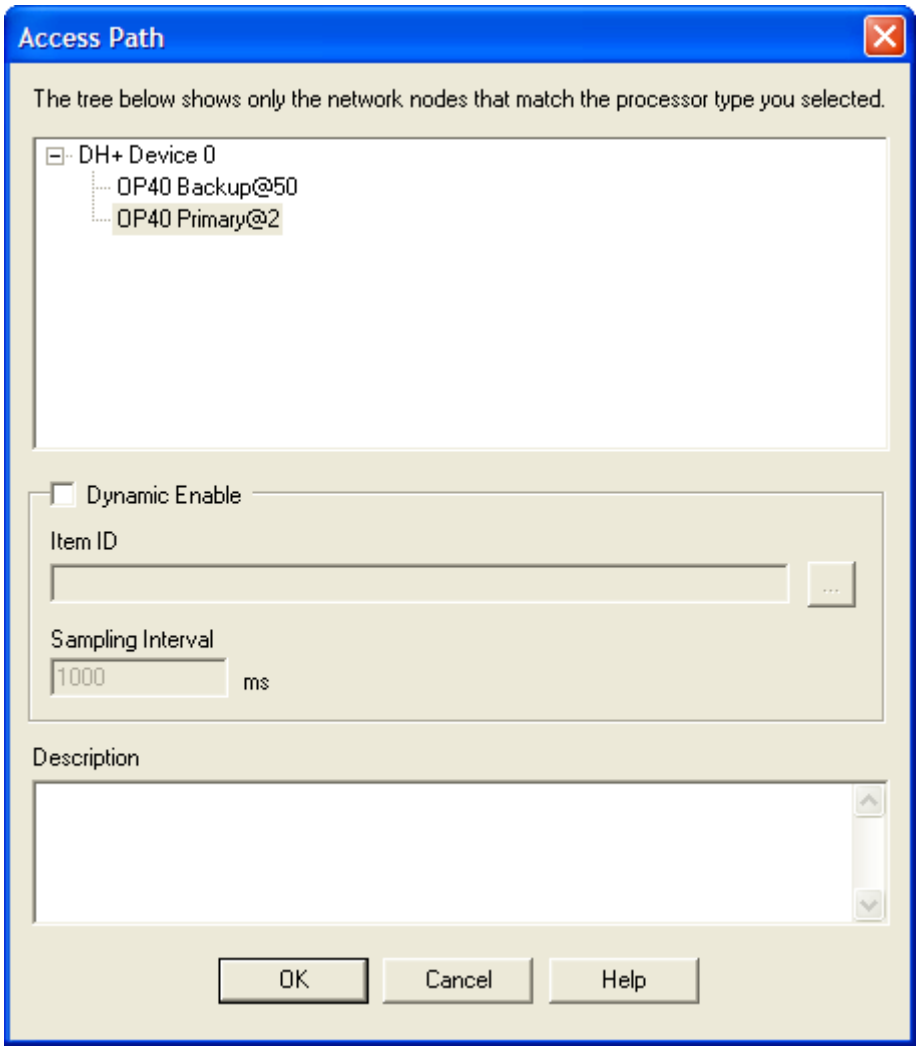
Information

If dynamic enable is configured for the highlighted access path, the enable data item will be shown here. In addition, if a description was entered for the access path, it will be displayed here.

This information is edited on the Access Path dialog box, described below.

New...

Click the ***New...*** button (or right-click inside the list window and select ***New...*** from the context menu) to create a new access path. The Access Path dialog box opens.



Select the network node for this access path.

If you check the Dynamic Enable box, you can specify an Item ID that will be used to control the enable status of the access path. Enter a data item or DirectAccess item ID in the box, or click the browse button to the right of it to browse for the desired item. You can use a Math & Logic item, if you wish. The Sampling Interval allows you to specify how often the enable item should be checked.

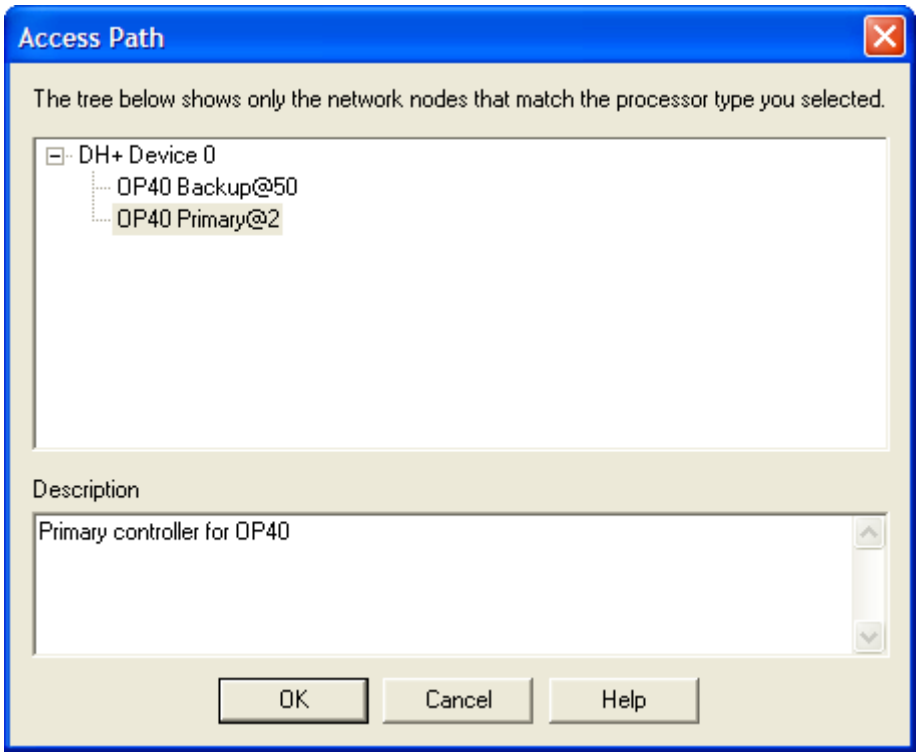
If the value of the enable item is false (Boolean false or a register value of 0), then the access path will be disabled. If the value is true (Boolean true or a nonzero register value), then the access path will be enabled.

You can also enter an optional description text of up to 255 characters.

Click ***OK*** when you are done.

Edit...

To modify an existing access path, select it and click the ***Edit...*** button (or right-click on the access path and select ***Edit...*** from the context menu). The Access Path dialog box opens.



Modify the current selections and click ***OK*** when you are done.

Delete

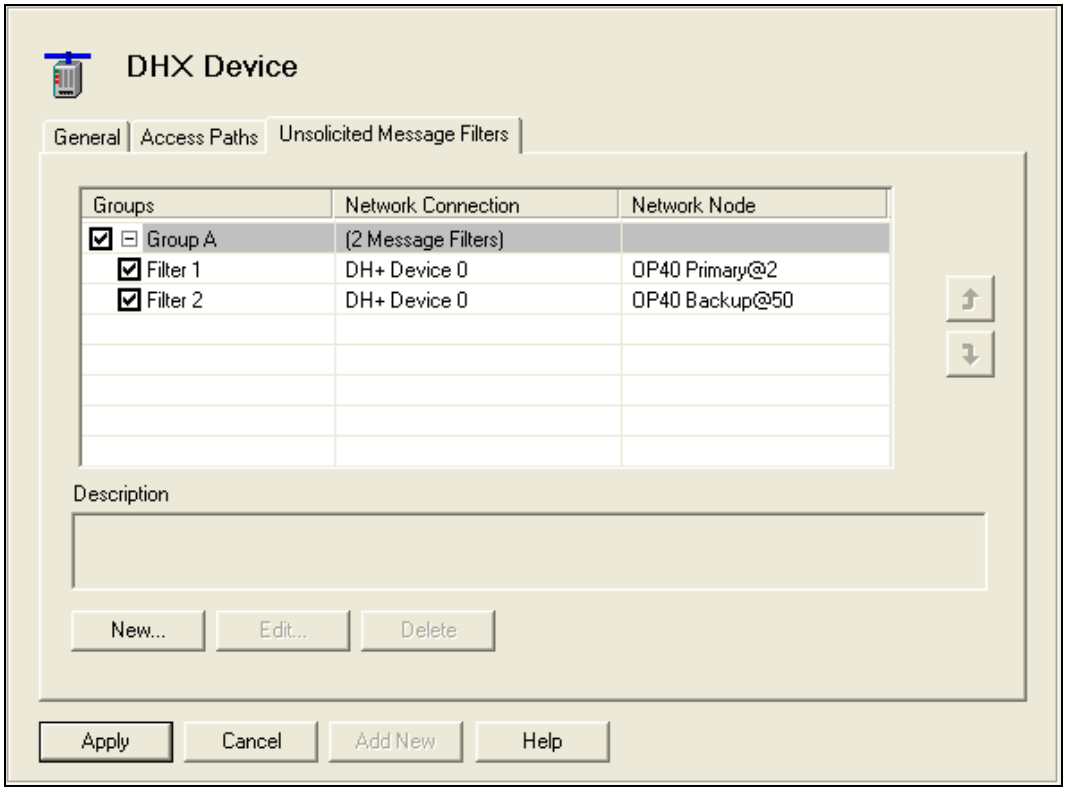
To delete an existing access path, select it and click the ***Delete*** button (or right-click and select ***Delete*** from the context menu).

Unsolicited Message Filters Tab

If the Accept All Unsolicited box on the General tab is checked, the server will accept all unsolicited messages. Otherwise, unsolicited messages must pass the filter criteria to be accepted. These filters help ensure that unsolicited messages are accepted only from trusted sources.

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 in order to be accepted by the server.

An unsolicited message filter group has a list of trusted network nodes and/or trusted network connections, and supports two modes of operation. In the default non-priority mode, the server will accept messages that pass any of the configured filters. In the alternative, priority 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.



Group

You can arrange the filters in one or more groups. Each group can be configured independently as either a prioritized or a non-prioritized list. There is no implied priority from one group to another.

Enable Checkbox

To the left of each unsolicited message group or filter is a checkbox that, when checked, enables that group/filter. The server uses only the enabled groups and filters.

Network Connection

The Network Connection column displays the network connection associated with the filter.

Network Node

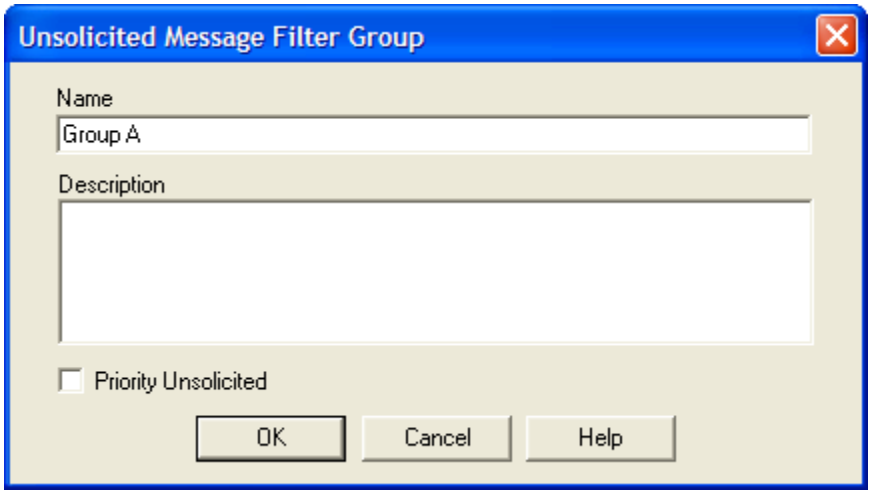
The Network Node column displays the network node associated with the filter.

Changing Filter Priority

The server assigns priority to the filters in a group from top to bottom, with the filter at the top having the highest priority. Use the up and down arrows on the right side of the list box to adjust the priorities as needed.

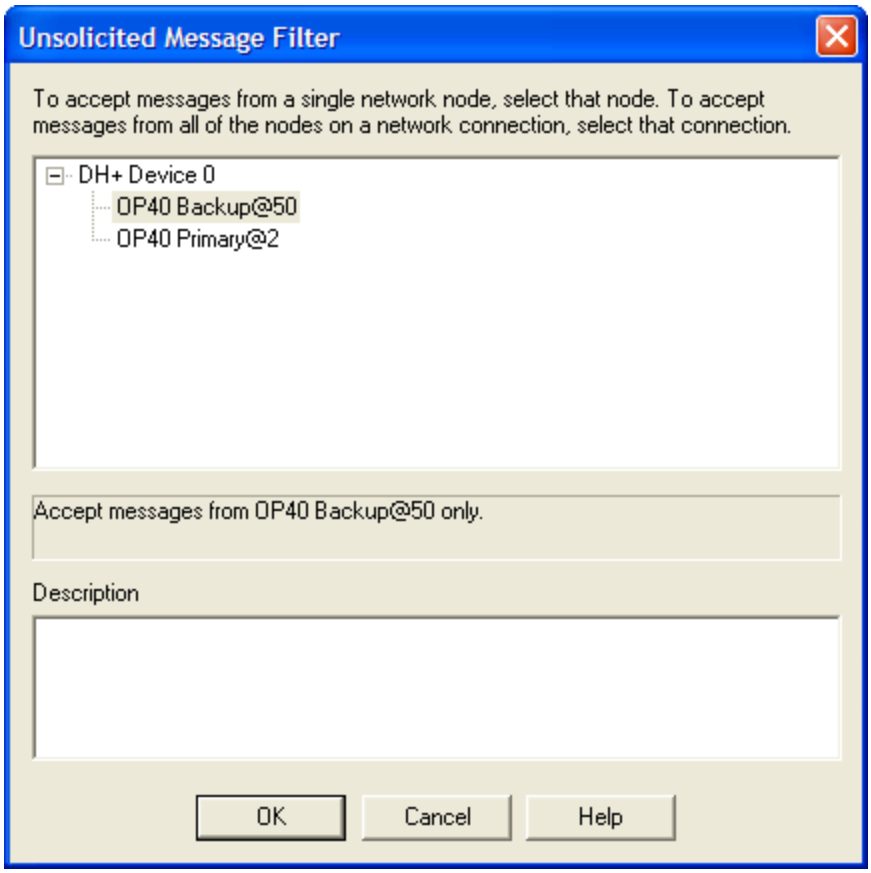
New...

To create a new group, click the ***New...*** button and select ***Group...*** (or right-click inside the list window and select ***New*** then ***Group...*** from the context menu). The Unsolicited Message Filter Group dialog will open.



Enter the ***Name*** of the group and an optional ***Description***, if desired. If you want this to be a prioritized list of filters, check the ***Priority Unsolicited*** box. Click ***OK*** when you are done.

To create a new filter, select the group in which you wish to create the filter. Click the ***New...*** button and select ***Filter...*** (or right-click on the desired group and select ***New...*** then ***Filter...*** from the context menu). The Unsolicited Message Filter dialog will open.



Select the network connection or network node you wish to use. You may also enter an optional ***Description***, if desired. Click the ***OK*** button when you are done.

Edit...

Select an existing group or filter and click the ***Edit...*** button (or right-click on an existing group or filter, and select ***Edit...*** from the context menu). The appropriate dialog box will. Modify the current selections and click ***OK*** when you are done.

Delete

Select an existing group or filter and click the ***Delete*** button (or right-click on the group or filter, and select ***Delete*** from the context menu).

Folders

A folder logically groups data items and other folders. You can place folders directly under devices or under other folders, up to four levels deep.

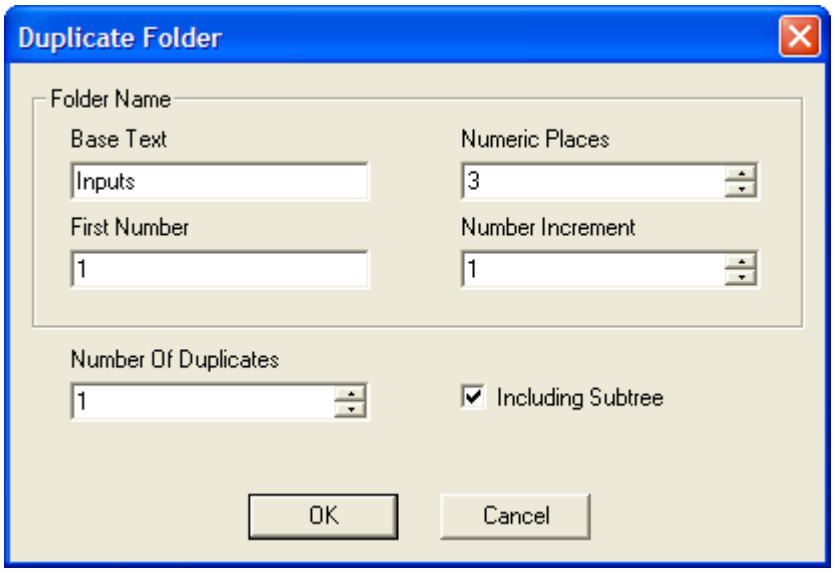
Caution! After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

Creating a New Folder

Right-click on an existing device or folder, and select ***New*** and then ***Folder*** from the context menu.

Duplicating a Folder

To speed up the creation of similarly-configured folders, you can create multiple folders in a single operation by duplicating an existing one. To do this, right-click on an existing folder and select ***Duplicate...*** from the context menu.



The above dialog box opens. You must specify how the duplicates are to be named by entering values for the ***Base Text***, ***First Number***, ***Numeric Places*** and ***Number Increment*** fields. To generate names for the duplicated folders, the editor begins with the base text and appends a number to it. The first duplicate uses the selected First Number value with the specified number of digits. This number is then incremented by the specified number for each of the remaining duplicates.

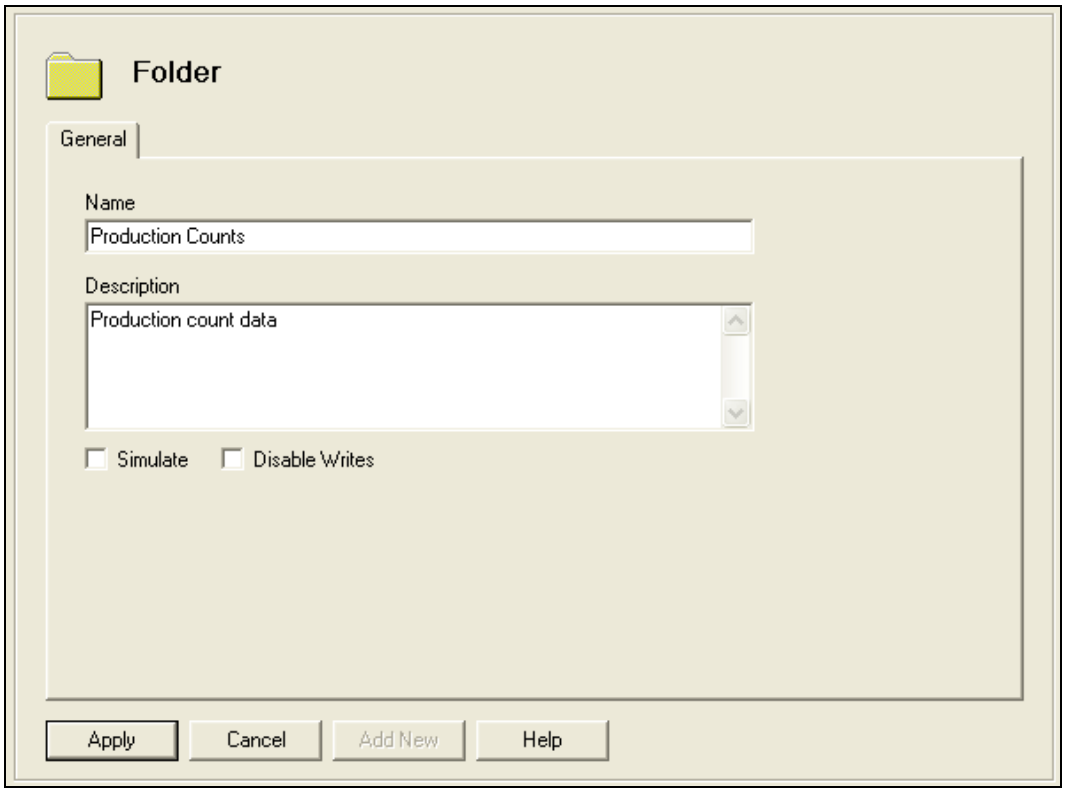
As an example, if Numeric Places is 3 and First Number is 2, the number 002 will be appended to the base text.

Use the ***Number Of Duplicates*** field to specify the number of folders you wish to create. If you want to duplicate all branches within the original folder, check the ***Including Subtree*** checkbox.

Deleting a Folder

To delete an existing folder, select it and press the ***Delete key***, or right-click on the folder and select ***Delete*** from the context menu.

General Tab



Name

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

Description

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

Simulate

Checking this box enables data simulation for all data items found at this level or below. This provides a quick way to switch between real and simulated data for a large number of data items. Refer to the Cyberlogic OPC Server Help for a full discussion about simulating data.

Note If the Simulate checkbox is grayed-out, it indicates that simulation has already been selected at a higher level.

Disable Writes

Checking this box disables write requests for all data items found at this level or below. By default, this box is not checked and writes are enabled.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled a higher level.

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 specific types of registers available vary from one PLC type to another, but all of the register types for all PLC models are available for configuration.

Caution! After you edit the configuration, you must open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

Creating a New Data Item

Right-click on an existing device or folder, and select ***New*** and then ***Data Item*** from the context menu.

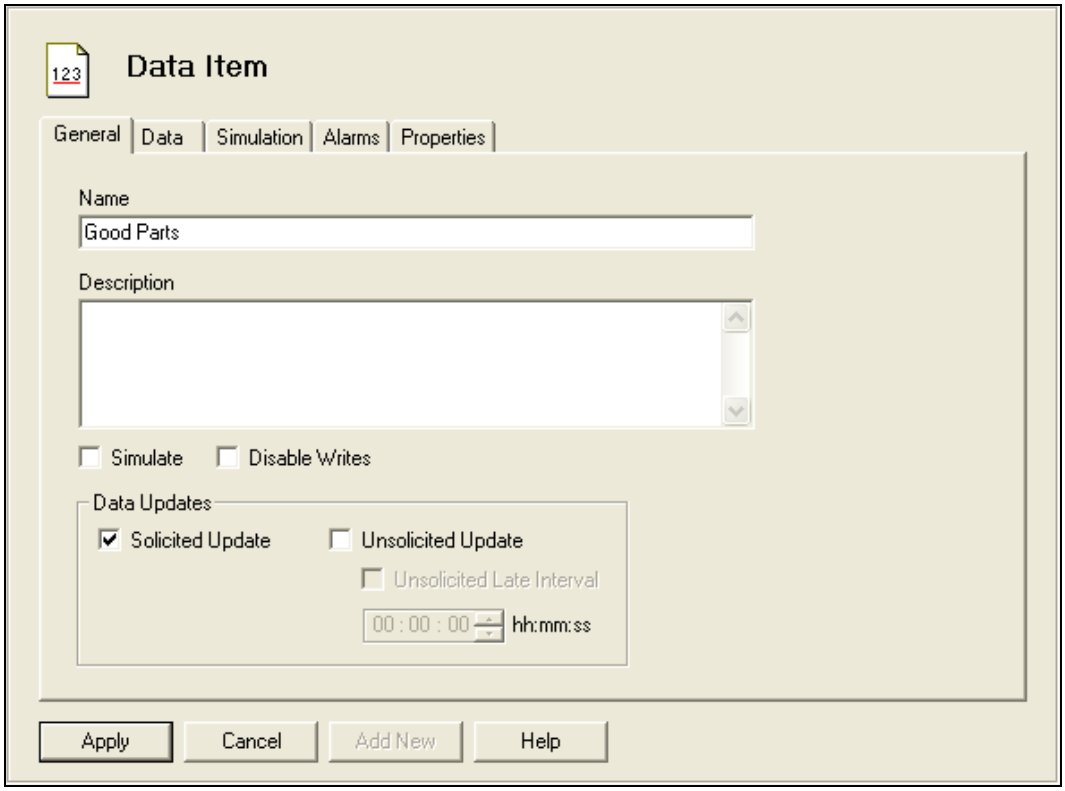
Duplicating a Data Item

To speed up the creation of similarly-configured data items, you can easily create multiple data items by duplicating an existing one. To do this, right-click on an existing data item and select ***Duplicate...*** from the context menu. The Data Item Duplication Wizard will open and will guide you through the duplication process. For details of this operation, refer to Appendix B: Data Item Duplication Wizard.

Deleting a Data Item

To delete an existing data item, select it and press the ***Delete key***, or right-click on the data item and select ***Delete*** from the context menu.

General Tab



Name

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

Description

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

Simulate

Checking this box enables data simulation for this data item. Refer to the Cyberlogic OPC Server Help for a full discussion about simulating data.

Note If the Simulate checkbox is grayed-out, it indicates that simulation has already been selected at a higher level.

Disable Writes

Checking this box disables all write requests for this data item. By default, this box is not checked and writes are enabled.

Note If the Disable Writes checkbox is grayed-out, it indicates that writes have already been disabled a higher level.

Solicited Update

When checked, this box tells the server to update this data item by polling. The server will use the access paths specified in this item's parent device to retrieve that data.

Unsolicited Update

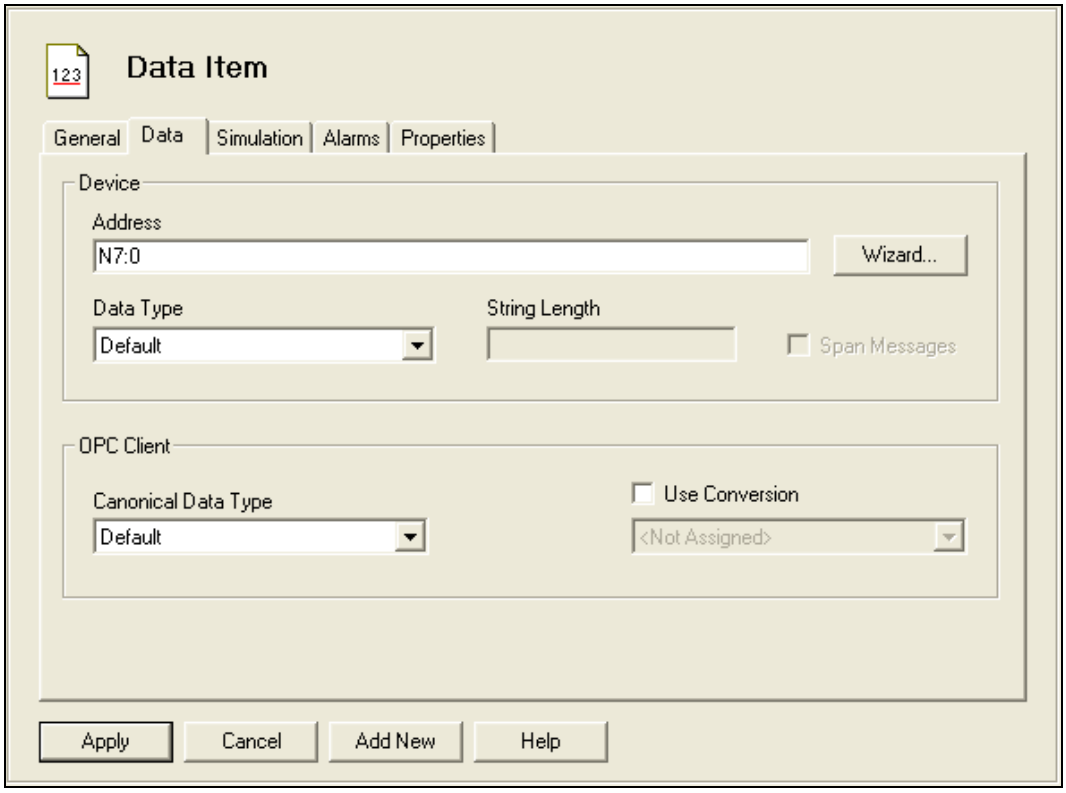
When checked, this box tells the server to allow this data item to be updated through unsolicited messages. The server will use the unsolicited message filters specified in this item's parent device to decide which unsolicited messages will be allowed to modify this value. This box is unchecked by default, disabling unsolicited updates.

Note Normally, you will not want to use both solicited and unsolicited updates at the same time on the same data item. Therefore, you should check one of these boxes and clear the other.

Unsolicited Late Interval

This is the maximum interval at which the server will expect to receive unsolicited updates for this data item. If the server does not receive an unsolicited update for this data item within this interval, the item's quality is downgraded to *Uncertain*.

Data Tab



Address

This is the address of the data in the physical device.

The valid syntax for this field depends upon the PLC family. In general, all addresses, as documented in Allen-Bradley manuals, are acceptable. However, the DHX Driver Agent extends this syntax when it comes to specifying arrays, array elements and bit fields. For a complete list of all valid addresses for various PLC types, refer to Appendix A: PLC Addresses.

Wizard...

If you have difficulties with specifying valid addresses, click this button and the Address Wizard will help you to define the address correctly.

For details on running the wizard, refer to Appendix C: Address Wizard.

Data Type

This drop-down allows you to select the native type of the data as it is found in the physical device. This tells the server how many bytes of data to read and write for this data item. Not all data types are allowed for all register types.

Normally, you will set the type to Default, which selects the usual data format for that type of register. However, in unusual applications where the data is stored in the physical

device in a non-standard way, you can specify a different format to simplify the access to the data.

String Length

If the data type is a string format, this field specifies the number of characters in the string. Remember that Unicode strings require two bytes per character.

Span Messages

This check box is available only for strings, arrays and bit fields.

If the box is not checked, the server will deliver all of the data for this data item in a single message. However, large data items may not fit into a single message, in which case the Span Messages box must be checked.

If the box is checked, the server may take more than a single message to obtain all data for this item. This may lead to data tearing if the data changes between the messages.

Canonical Data Type

This selection specifies the default variant format (VT_xxx) in which the data will be sent to OPC clients. Each OPC client can request data in any variant format and consequently override the Canonical Data Type setting.

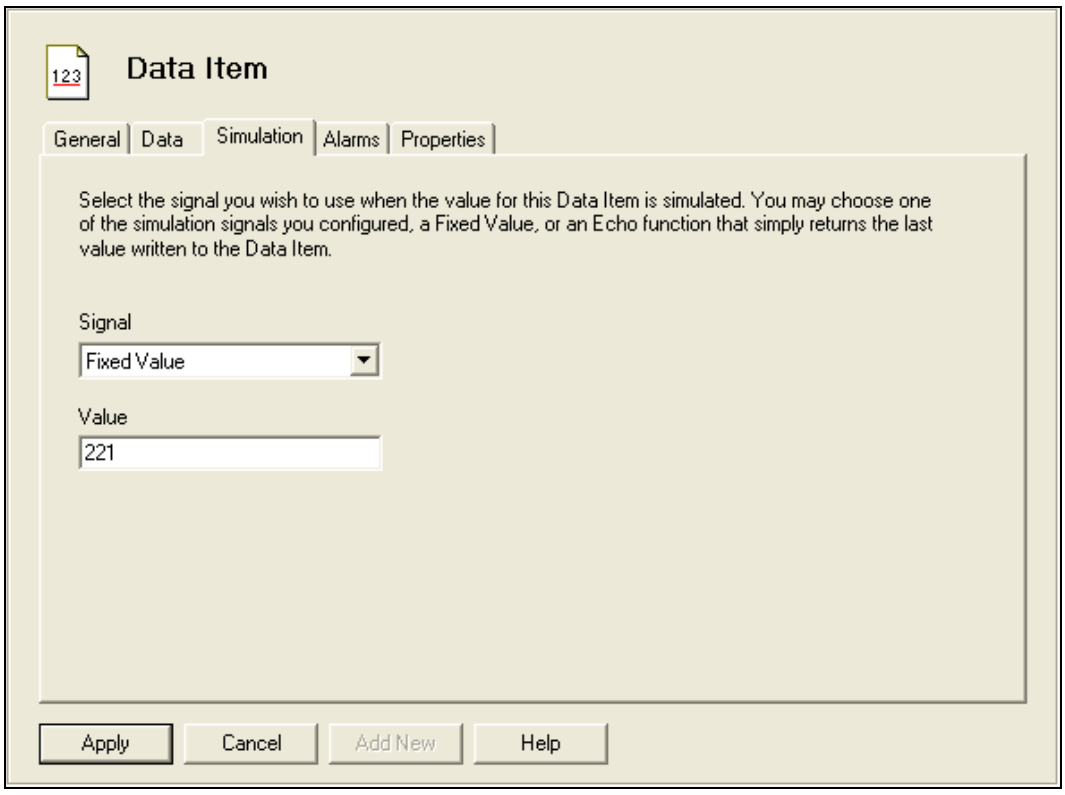
The most common selection is Default, in which case the server matches the variant type to the item's data type. The following table shows the default mappings.

Type	Size in bits	Default Canonical Data Type	.NET Data Type	Description
Default				Default type based on the data item address
BIT	1	VT_BOOL	bool	1-bit boolean
SINT8	8	VT_I1	sbyte	Signed 8-bit integer -128 to 127
UINT8	8	VT_UI1	byte	Unsigned 8-bit integer 0 to 255
SINT16	16	VT_I2	short	Signed 16-bit integer -32,768 to 32,767
UINT16	16	VT_UI2	ushort	Unsigned 16-bit integer 0 to 65,535
SINT32	32	VT_I4	int	Signed 32-bit integer -2,147,483,648 to 2,847,483,647
UINT32	32	VT_UI4	uint	Unsigned 32-bit integer 0 to 4,294,967,295
SINT64	64	VT_I8	long	Signed 64-bit integer −9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
UINT64	64	VT_UI8	ulong	Unsigned 64-bit integer 0 to 18,446,744,073,709,551,615
FLOAT32	32	VT_R4	float	IEEE 32-bit floating point number ±3.4e±38
FLOAT64	64	VT_R8	double	IEEE 64-bit floating point number ±1.7e±308
BCD16	16	VT_UI2	ushort	BCD value; 0 to 9,999
BCD32	32	VT_UI4	uint	BCD value; 0 to 99,999,999
STRING	String size * 8	VT_BSTR	string	Zero terminated ASCII string of 8-bit characters
WSTRING	String size * 16	VT_BSTR	string	Zero terminated UNICODE string of 16-bit characters
FIELD	Field size	Best fitting VT_UIx or array of VT_UI1 if size > 64	Best fitting unsigned type or byte[ ] if size > 64	Multiple bit field

Use Conversion

Check this box to indicate that the selected conversion should be applied to the data before the value is stored in the data item cache. You must select one of the previously-configured Conversions from the drop-down box.

Simulation Tab



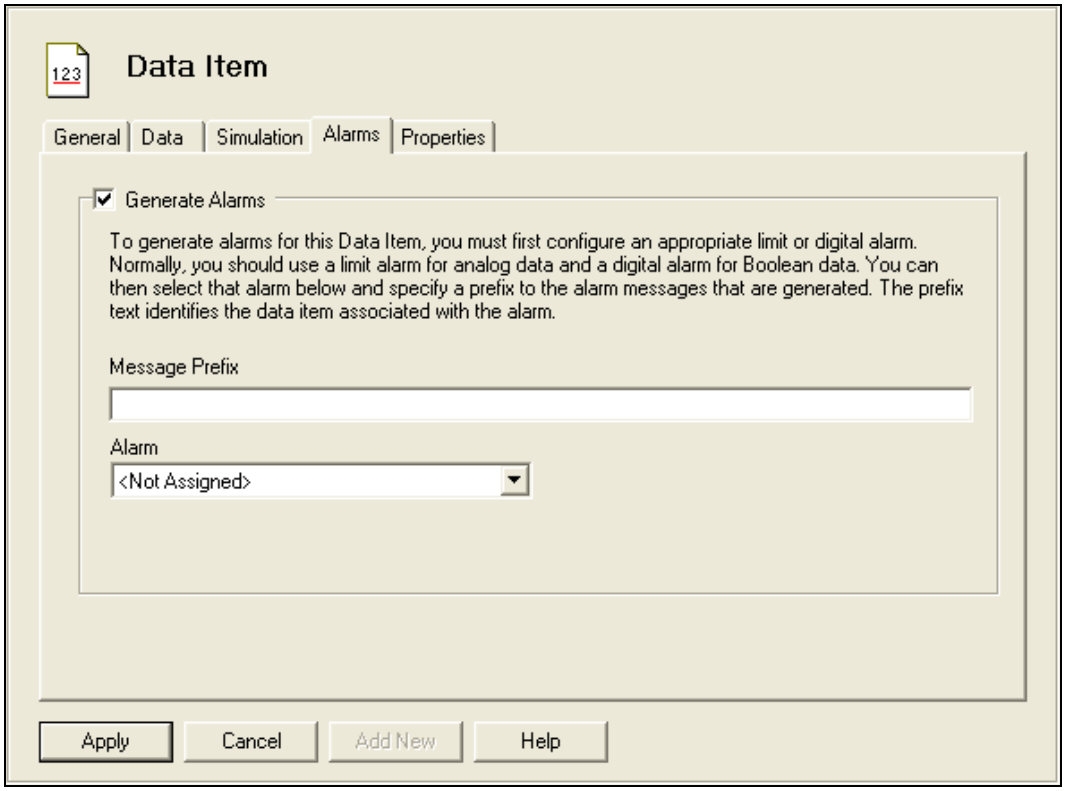
Signal

If you enabled simulation on the General tab or at a higher level, you may choose to simulate the data item value with one of the previously-defined Simulation Signals, a fixed value or an echo of the last value written to the item.

Value

When simulation is enabled and the Signal field is set to Fixed Value, the data item will be set to this value.

Alarms Tab



Generate Alarms

If this box is checked, the server will test the alarm conditions for this data item, generating alarms as appropriate.

Refer to the Cyberlogic OPC Server Help for a full discussion on creating and using alarms.

Message Prefix

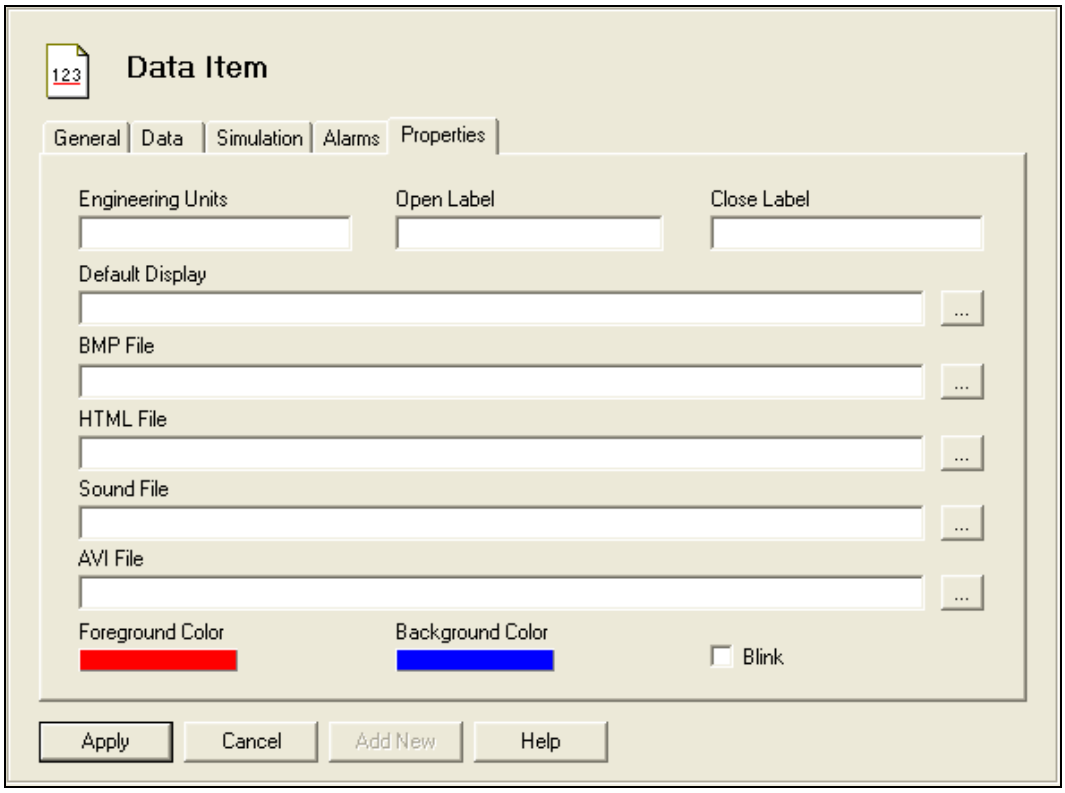
Enter the text for the first part of the alarm message. The second part will be the body text of the specific alarm that is generated.

Alarm

Select one of the previously-defined Alarm Definitions to serve as the alarm template for this data item.

Properties Tab

In addition to the main data item properties—value, quality and timestamp—the OPC specification includes several optional properties that your client application may use. This tab allows you to set these data item properties. These properties are static and do not change while the server is running.



Engineering Units

This is OPC property ID 100. It specifies the engineering units text, such as DEGC or GALLONS. It can be up to 50 characters long.

Open Label

This is OPC property ID 107, and is presented only for discrete data. This text describes the contact when it is in the open (zero) state, such as STOP, OPEN, DISABLE or UNSAFE. It can be up to 50 characters long.

Close Label

This is OPC property ID 106, and is presented only for discrete data. This text describes the contact when it is in the closed (non-zero) state, such as RUN, CLOSE, ENABLE or SAFE. It can be up to 50 characters long.

Default Display

This is OPC property ID 200. It is the name of an operator display associated with this data item. It can be up to 255 characters long.

BMP File

This is OPC property ID 204. It is the name of a bitmap file associated with this data item, for example C:\MEDIA\FIC101.BMP. It can be up to 255 characters long.

HTML File

This is OPC property ID 206. It is the name of the HTML file associated with this data item, for example http://mypage.com/FIC101.HTML. It can be up to 255 characters long.

Sound File

This is OPC property ID 205. It is the name of the sound file associated with this data item, for example C:\MEDIA\FIC101.WAV. It can be up to 255 characters long.

AVI File

This is OPC property ID 207. It is the name of the AVI file associated with this data item, for example C:\MEDIA\FIC101.AVI. It can be up to 255 characters long.

Foreground Color

This is OPC property ID 201. Click on the box and select the foreground color used to display the item.

Background Color

This is OPC property ID 202. Click on the box and select the background color used to display the item.

Blink

This is OPC property ID 203. Check this box to indicate that displays of the item should blink.

Conversions

The raw data associated with data items may be process values from instruments. 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 conversions. A number of data items can then use each conversion. As a result, the user need not define the same conversion many times over.

Refer to the Cyberlogic OPC Server Help for a full discussion.

Simulation Signals

The server can simulate the data for each of the data items 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

Each signal has parameters that define properties such as amplitude, phase and number of steps.

Refer to the Cyberlogic OPC Server Help for a full discussion.

Alarm Definitions

The Cyberlogic OPC Server supports the OPC Alarms and Events specification, permitting it to generate alarms based on the value of data items.

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 with Boolean data items and limit alarms are normally used with numeric data items, but both types of alarms may be used with either data type. Alarms cannot be used with string or array data items or with bit fields larger than 64 bits.

Refer to the Cyberlogic OPC Server Help for a full discussion.

Note Configuring alarms is meaningful only if your client software also supports the OPC Alarms & Events specification. Consult your client software documentation to see what specifications it supports.

Database Operations

In addition to providing data to OPC clients in real time, the Cyberlogic OPC Server can store it in a database. The feature that does this is called Data Logger. Once the data is logged, it is available to any application that can access that database. It need not be an OPC client application.

Refer to the Data Logger Help for a full discussion.

OPC Crosslinks

OPC Crosslinks allow you to transfer data from an OPC server or PLC to other OPC servers or PLCs. The data item you read from is called the crosslink input. You may write its value to any number of data items, and these are called crosslink outputs.

Refer to the OPC Crosslink Help for a full discussion.

Saving and Undoing 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***, or click the ***Save & Update Server*** toolbar button, for the changes you have made to take effect. Otherwise, the server will still be running with the old configuration.

Saving Configuration Changes Without Updating the Server

To save the configuration without updating the server, open the ***File*** menu and select ***Save***, or click the ***Save*** button on the toolbar. The changes will be saved but the server will still be running with the old configuration.

Saving Configuration Changes and Updating Server

To save the configuration and update the server, open the ***File*** menu and select ***Save & Update Server***, or click the ***Save & Update Server*** button on the toolbar.

Undoing Configuration Changes

To undo configuration changes and revert to the previously saved configuration, open the ***File*** menu and select ***Undo Changes***, or click the ***Undo Changes*** button on the toolbar.

Configuration Import/Export

The Import/Export feature allows you to export the configuration data to text file format and import configuration data from these exported files and also from comma separated values files from other vendors' OPC servers and programming software.

For details on this important feature and instructions in its operation, refer to the Cyberlogic OPC Server Help.

Options

The editor has several options that may be set to adjust the operation of the editor to suit your preferences and to set security levels as needed for communication with client software. For a full discussion, refer to the Cyberlogic OPC Server Help.

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 DHX Driver Agent Messages and Frequently Asked Questions to assist in your troubleshooting.

Data Monitor

The Data Monitor lets you monitor the values and status of the data items. Its use is described in detail in the Cyberlogic OPC Server Help.

Cyberlogic OPC Client

The Cyberlogic OPC Client is a simple OPC Data Access client that lets you see how the server interacts with a client and lets you test its response to various loads. Its use is described in detail in the Cyberlogic OPC Server Help.

Performance Monitor

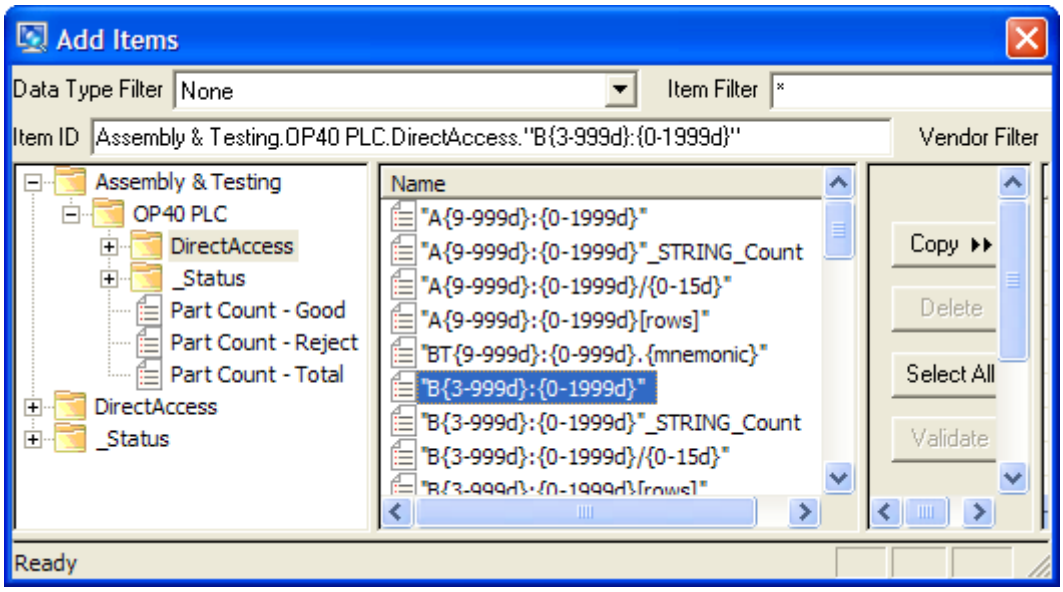
The Performance Monitor is a Microsoft diagnostic tool that the Cyberlogic drivers support. Its use is described in detail in the Cyberlogic OPC Server Help.

DirectAccess

At run time, in addition to the user-configured branches, the Cyberlogic OPC Server dynamically creates DirectAccess branches in its address space. These are created for both network nodes and devices.

DirectAccess allows read and write operations. However, for extra security, write operations are disabled by default. If writes are permissible, they can be enabled on a node-by-node basis as part of the network node configuration, and on a device-by-device basis as part of the device configuration.

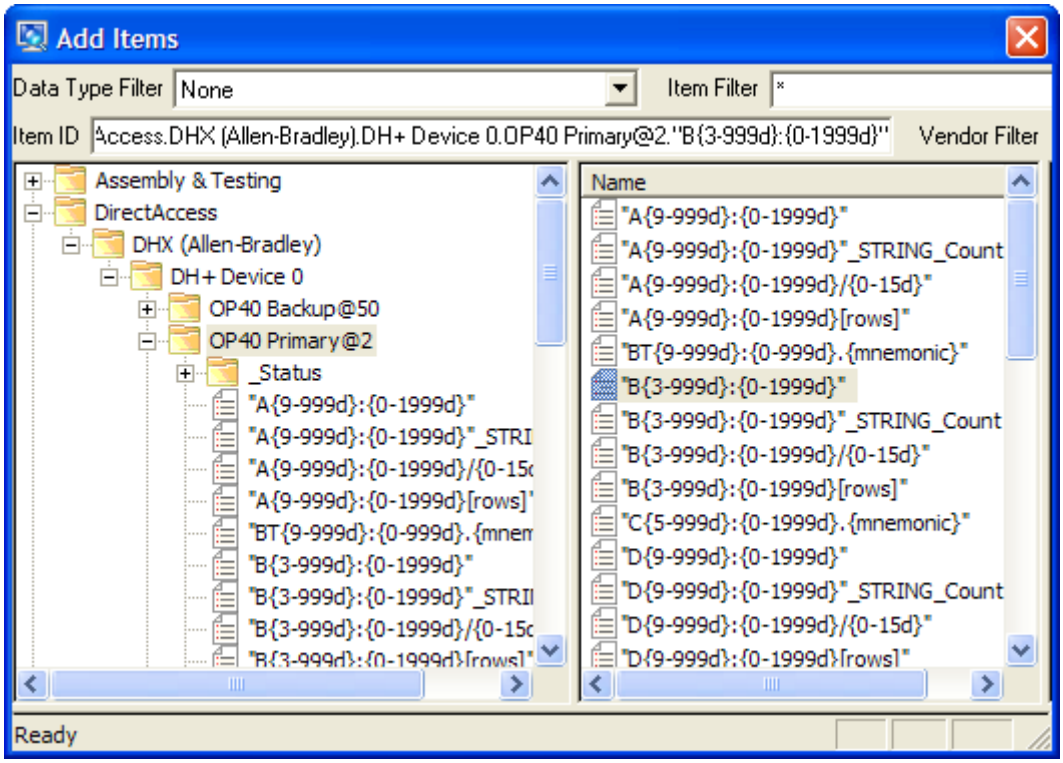
Device DirectAccess



Each device in the address space will contain all of its configured data items, plus a DirectAccess branch, as shown in the example above. This branch will appear only for devices that have DirectAccess enabled. OPC clients can then use this branch to access any register in the device by directly specifying the register address.

In the DirectAccess branch for a device, the Cyberlogic OPC Server reports a list of hints about the types of data items that may exist on the selected device. These are not valid item addresses. Rather, they are just hints to help the user to specify a proper address. For more details on using these hints, refer to the Address Hints section.

Network Node DirectAccess



DirectAccess to network nodes is achieved through a branch called DirectAccess at the root of the address space. This branch acts like a device folder that contains all of the configured network connections.

As you can see in the example above, 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 bit or register in any configured network node by directly specifying its 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 to help the user to specify a proper address.

Address Hints

In the PLC-5 example above, "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 using DirectAccess to the network node, you would edit the Item ID field at the top of the dialog box to read:

DirectAccess.DHX (Allen-Bradley).DH+ Device 0.OP40 Primary@2."B3:100"

To access the same register using DirectAccess to the device, you would edit the Item ID to read:

Assembly & Testing.OP40 PLC.DirectAccess."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".

Here are some additional examples of valid DirectAccess item addresses for a PLC-5:

"B3:100/1,5"

Bit field of five bits starting from bit B3:100/1

"N7:0[5]"

Array of five registers starting from address N7:0

"N7:0"_UINT16

Unsigned 16-bit integer at address N7:0. (The _UINT16 portion is a data type override.)

"ST20:0"

ASCII string at address ST20:0

"A16:10"_STRING_10

10-character ASCII string at address A16:10. (The _STRING_10 portion is a data type override.)

Here is a brief explanation of the other address hints:

{Any valid reference}#{Conversion}

This form allows you to apply a previously configured data conversion to a raw register value in order to convert it into a form that is more useful to the client. When the conversion name is preceded by a # sign, the canonical data type for the requested data will always be VT_R8 (64-bit floating point).

{Any valid reference}@{Conversion}

This form allows you to apply a previously configured data conversion to a raw register value in order to convert it into a form that is more useful to the client. When the conversion name is preceded by an @ sign, the canonical data type will match the data type of the requested register or the specified {Data Type Override}.

Note

The address hints are shown enclosed in double-quotes, and the item address you specify in place of the hint must also be enclosed in double-quotes. If a data type override is used, it is not enclosed in the double-quotes.

Previous versions of the Cyberlogic OPC Server did not require the double-quotes, but had the requirement that any periods (.) in the address had to be replaced with a forward slash (/). This format is still valid, for compatibility with existing configurations. However, the double-quote format is preferred for new configurations.

DirectAccess Address Formats

The listed hints cover only the most common address formats. In fact, any item address in one of the following formats is valid:

- {Register Address}
- {Register Address}_{Data Type Override}
- {Register Address}#{Conversion}
- {Register Address}_{Data Type Override}#{Conversion}
- {Register Address}@{Conversion}
- {Register Address}_{Data Type Override}@{Conversion}

Register Address

The Register Address portion may be any address that is acceptable in the Address field of a data item. This portion is enclosed in double-quotes.

The PLC addressing requirements are discussed in Appendix A: PLC Addresses.

Data Type Override

This capability permits you to display the value in a format other than the native format of the register. For example, a register might normally contain binary data, but is being used to hold BCD data for a particular application. You could then specify that it should be treated as BCD. The data type override portion is optional, and if used, is not enclosed in double-quotes.

The STRING and WSTRING data types may require the count value, which specifies the maximum number of characters in a string. By default, the count value is 1.

Any data type that is acceptable in the Data Type field of a data item may be specified as the override type. Notice that not all data types are valid for all register addresses. The table below shows all supported data types.

Caution!

The Data Type Override field requires you to use the form in the Type column, not the Canonical Data Type. For example, if you want 32-bit floating point format, you must specify ***FLOAT32***. The canonical form ***VT_R4*** will not work.

Type	Size in bits	Default Canonical Data Type	.NET Data Type	Description
Default				Default type based on the data item address
BIT	1	VT_BOOL	bool	1-bit boolean
SINT8	8	VT_I1	sbyte	Signed 8-bit integer -128 to 127
UINT8	8	VT_UI1	byte	Unsigned 8-bit integer 0 to 255
SINT16	16	VT_I2	short	Signed 16-bit integer -32,768 to 32,767
UINT16	16	VT_UI2	ushort	Unsigned 16-bit integer 0 to 65,535
SINT32	32	VT_I4	int	Signed 32-bit integer -2,147,483,648 to 2,847,483,647
UINT32	32	VT_UI4	uint	Unsigned 32-bit integer 0 to 4,294,967,295
SINT64	64	VT_I8	long	Signed 64-bit integer −9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
UINT64	64	VT_UI8	ulong	Unsigned 64-bit integer 0 to 18,446,744,073,709,551,615
FLOAT32	32	VT_R4	float	IEEE 32-bit floating point number ±3.4e±38
FLOAT64	64	VT_R8	double	IEEE 64-bit floating point number ±1.7e±308
BCD16	16	VT_UI2	ushort	BCD value; 0 to 9,999
BCD32	32	VT_UI4	uint	BCD value; 0 to 99,999,999
STRING	String size * 8	VT_BSTR	string	Zero terminated ASCII string of 8-bit characters
WSTRING	String size * 16	VT_BSTR	string	Zero terminated UNICODE string of 16-bit characters
FIELD	Field size	Best fitting VT_UIx or array of VT_UI1 if size > 64	Best fitting unsigned type or byte[ ] if size > 64	Multiple bit field

Conversion

This feature allows you to apply a previously configured data conversion to a raw register value in order to convert it into a form that is more useful to the client. For example, you can change a transducer's voltage value into a pressure value in psi.

The conversion name must be preceded by either a # sign or an @ sign. With the # sign, the canonical data type for the requested data will always be VT_R8 (64-bit floating point). With the @ sign, the canonical data type will match the data type of the requested register or the specified {Data Type Override}. Here are a couple of examples:

DirectAccess.DHX (Allen-Bradley).DH+ Device 0.OP40 Primary@2."B3:100"#100%
Assembly & Testing.OP40 PLC.DirectAccess."B3:100"_SINT16@P200

In the first example, the canonical data type is VT_R8, while in the second example, it is VT_I2. The *100%* and *P200* are the names of the conversions.

Event Viewer

During startup and operation, the Cyberlogic OPC Server may detect problems or other significant events. When a noteworthy event is detected, the server sends an appropriate message to the Windows Event Logger. You can view these messages using the Windows Event Viewer. Its use is described in detail in the Cyberlogic OPC Server Help.

For an explanation of the error messages that may be logged by the DHX Driver Agent, refer to the DHX Driver Agent Messages section.

DHX Driver Agent Messages

This section shows Error Log messages that may be generated by the DHX Driver Agent (DhxDriverClassSrv in the Source column). The main Cyberlogic OPC Server module can also log error messages (CybOpcRuntime in the Source column). For a list of these messages, refer to the Cyberlogic OPC Server Help.

Registration DLL failed to load. The I/O operations of the server have been disabled. Reinstall the product.

The installation program should have copied this DLL into the Windows system32 directory. Reinstall the product.

Registration verification failed. The I/O operations of the server have been disabled. Reinstall the product.

The server could not access the registration information, which is gathered and stored during the installation process. Reinstall the product.

This is a <hours>-hour promotional copy of the DHX driver agent. The Cyberlogic OPC Server started at <start time> and the agent will stop at <stop time>.

This is a time-limited version of the Cyberlogic OPC Server. For information on upgrading to the full version, contact your Cyberlogic distributor or visit Cyberlogic's website at www.cyberlogic.com.

This is a promotional copy of the DHX OPC Server. The allowed operation time has expired. The I/O operations of the server have been disabled.

This is a time-limited version of the DHX OPC Server and the allowed operation time has expired. For information on upgrading to the full version, contact your Cyberlogic Software, Inc. distributor or visit Cyberlogic's website at www.cyberlogic.com.

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.

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.

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.

During device type detection, network node device "<network node name>" reported an unrecognized device ID (<device id>). Automatic device type detection cannot be used with this device. Defaults to <value>. Report this message to the manufacturer's technical support.

The OPC server software tried to identify the specified network node, but it reported a device ID that the editor does not recognize. This may indicate a new device type that has not yet been added to the editor's device type database. The server configuration has defaulted to the ID value given in the message. You should check to see if this is an acceptable default, and manually change it if it is not. Please report this error to Cyberlogic's Tech Support so we can include this device type in future releases of the software.

During device type detection, network node device "<network node name>" reported an unrecognized device ID (<device id>). Automatic device type detection cannot be used with this device. Report this message to the manufacturer's technical support.

The OPC server software tried to identify the specified network node, but it reported a device ID that the editor does not recognize. This may indicate a new device type that has not yet been added to the editor's device type database. You must edit the network node and select the proper device type. Please report this error to Cyberlogic's Tech Support so we can include this device type in future releases of the software.

The Cyberlogic License Server failed to respond with valid license information. The I/O operations of the DHX driver agent have been disabled. Contact the manufacturer's technical support.

The driver agent experienced a problem when it tried to contact the Cyberlogic License Server. If the license server is not running, start it and then try restarting the OPC server. If the license server is already running, contact Cyberlogic Tech Support.

Frequently Asked Questions

For FAQs common to all driver agents refer to the Cyberlogic OPC Server Help.

I have a 1784-PKTX card connected to the Data Highway Plus network. I successfully installed the DHX OPC Server, but when I try the auto configuration, the editor fails to detect any network connections. What is the problem?

For the auto configuration to work you must first create at least one DHX device. To do that, at least one of the following DHX family drivers must be installed: DHX Driver, Ethernet DHX Driver, Serial DHX Driver or DHX Gateway Driver. (In your case, you should install at least the DHX Driver.) Refer to the driver-specific help file for information on creating DHX devices. Be sure the driver is running. The drivers default to automatic operation when they are installed, but may have been switched to manual. This selection is done from the driver configuration editor.

I have a Plug and Play adapter card for Data Highway (or Data Highway Plus)

installed in my system. There is no communication and the DHX demo software doesn't show any of the existing nodes.

The Plug and Play cards for Allen-Bradley networks all default to node address 0. If there is another one of these cards anywhere on your network, it is likely that it is using this default address, resulting in a conflict. Use the DHX Configuration Editor to change the adapter's node address to a different value.

I have ControlLogix controllers on a Data Highway Plus network. When I try the auto configuration, the editor fails to detect any of them, but does "detect" a bunch of nodes on the DHX Network Connections for Allen-Bradley branch and shows them as controllers of "Unknown" type. What is the problem?

ControlLogix nodes will not show up under the ControlLogix Network Connections branch because the ControlLogix driver agent does not support the auto configuration feature.

The DHX driver agent will detect the nodes, but because ControlLogix processors are not directly supported by that driver agent, it will show them as "Unknown". However, you can configure PLC-5 compatibility files inside the ControlLogix unit that can be used for PLC-5 style communications. Refer to Appendix E: Using ControlLogix Processors for more information.

APPENDIX A: PLC ADDRESSES

The allowed syntax for the Address field on the Data Tab of a data item depends upon the PLC type. In general, the Cyberlogic OPC Server supports the standard naming conventions as used by Allen-Bradley. These are described in detail for the various controller types:

- PLC-2
- PLC-3
- PLC-5 Family
- PLC-5/10, 5/12, 5/15, 5/25
- SLC 5/04, 5/05
- SLC-500 Fixed I/O, SLC 5/01, 5/02, 5/03
- MicroLogix

However, the DHX Driver Agent extends this syntax when it comes to specifying arrays, array elements and bit fields.

Arrays

Single-dimensional arrays are supported for data types other than strings and bit fields. To specify an array of data type elements, use the following syntax:

{Valid Register Address}**[**{Number of Elements}**]**

or

{Valid Register Address}**[**{Number of Elements}**,**{Lower Bound}**]**

The Lower Bound specifies the array index value for the first element in an array. If not specified, the Lower Bound defaults to zero. Visual Basic applications may expect the first index to be one, in which case you would set the Lower Bound value to 1. Here are a few examples of arrays for a PLC-5:

N7:0[5]	Array of five registers starting with N7:0, with a lower bound of 0
O:0[4,1]	Array of four registers starting with O:0, with a lower bound of 1
I:10/0[16]	Array of sixteen bits starting with I:10/0, with a lower bound of 0

Array Elements

Some complex data types, such as the PLC-5 PID control structure, may contain built-in arrays of a certain type. To specify an element of such array, use the following syntax:

{Valid Register Address}**(**{ Element Index}**)**

Here are a few examples of array elements for a PLC-5:

PD18:1.ADDR(0) First element of PD18:1.ADDR array

A16:1(0) First byte of register A16:1

A16:1(1) Second byte of register A16:1

Notice that it is possible to specify whether you want byte 0 or byte 1 of an ASCII register. It is also possible to combine this specification with the array syntax to specify an array of ASCIIs beginning at a specific byte. For example:

A16:1(1)[10] Array of 10 ASCII characters starting from the second byte of A16:1

In the case of PLC-3 High Order Integer files, you can use this same syntax to specify which 16-bit word you wish to address.

H23:3(0) First word of register H23:3

H23:3(1) Second word of register H23:3

H23:3(1)[6] Array of six 16-bit words beginning with the second word of register H23:3

Bit Fields

To specify a sequence of bits as a bit field, use the following syntax:

{Valid Bit Address}**,**{Bit Count}

Here are a few examples of bit fields for a PLC-5:

B5:100/1,5 Bit field of five bits starting from bit B5:100/1

I:10/0,16 Bit field of sixteen bits starting with I:10/0,with a lower bound of 0

PLC-2

PLC-2s do not support file types. They simply address memory by words and optionally by bits. They support counters and timers, but these are not addressed as a structure. You must know where the timer is located, read the word or bit and interpret it accordingly.

All Data

Syntax	Ranges
{word}	{0-17777o}
{word}/{bit}	{0-17777o}/{0-17o}
{word}/{bit},{cnt}	{0-17777o}/{0-17o},{1-16d}
{word}.{mnemonic}	{0-17777o}.{mnem}
{word}.{mnemonic}/{bit}	{0-17777o}.{mnem}/{0-17o}
{word}.{mnemonic}/{bit},{cnt}	{0-17777o}.{mnem}/{0-17o},{1-16d}
{word}/{mnemonic}	{0-17777o}/{mnem}
{word}/{mnemonic}/{bit}	{0-17777o}/{mnem}/{0-17o}
{word}/{mnemonic}/{bit},{cnt}	{0-17777o}/{mnem}/{0-17o},{1-16d}

The default data type for PLC-2 controllers is three-digit binary coded decimal (BCD). Arrays are supported.

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): DN, EN, CU, CD, OU

PLC-3

Input Files

Syntax	Ranges
I{file}:{word}	I{0-999d}:{0-7777o}
I{file}:{word}/{bit}	I{0-999d}:{0-7777o}/{0-17o}
I{file}/{bit}	I{0-999d}:{0-177777o}
I:{word}	I:{0-7777o}
I:{word}/{bit}	I:{0-7777o}/{0-17o}
I/{bit}	I/{0-177777o}

If no file is specified, file zero (0) is assumed.

Output Files

Syntax	Ranges
O{file}:{word}	O{0-999d}:{0-7777o}
O{file}:{word}/{bit}	O{0-999d}:{0-7777o}/{0-17o}
O{file}/{bit}	O{0-999d}:{0-177777o}
O:{word}	O:{0-7777o}
O:{word}/{bit}	O:{0-7777o}/{0-17o}
O/{bit}	O/{0-177777o}

If no file is specified, file zero (0) is assumed.

Status Files

Syntax	Ranges
S{file}:{word}	S{0-999d}:{0-9999d}
S{file}:{word}/{bit}	S{0-999d}:{0-9999d}/{0-17o}
S{file}/{bit}	S{0-999d}/{0-470377o}

Arrays are supported.

ASCII Files

Syntax	Ranges
A{file}:{word}	A{0-999d}:{0-9999d}
A{file}:{word}/{bit}	A{0-999d}:{0-9999d}/{0-17o}
A{file}/{bit}	A{0-999d}/{0-470377o}

Arrays are supported.

Binary Files

Syntax	Ranges
B{file}:{word}	B{0-999d}:{0-9999d}
B{file}:{word}/{bit}	B{0-999d}:{0-9999d}/{0-17o}
B{file}/{bit}	B{0-999d}/{0-470377o}

Arrays are supported.

Timer Files

Syntax	Ranges
T{mnemonic}:{word}/{bit}	T{mnem}:{0-9999d}/{0-17o}
T{mnemonic}{word}/{bit}	T{mnem}{0-9999d}/{0-17o}
T{word}.{mnemonic}	T{0-9999d}.{mnem}/{0-17o}
T{word}/{mnemonic}	T{0-999d}/{mnem}/{0-17o}

Valid mnemonics

Word fields (read/write): ACC, PRE, CTL

Boolean fields (read only): TD, TE, TT

Counter Files

Syntax	Ranges
C{mnemonic}:{word}/{bit}	C{mnem}:{0-9999d}/{0-17o}
C{mnemonic}{word}/{bit}	C{mnem}{0-9999d}/{0-17o}
C{word}.{mnemonic}	C{0-9999d}.{mnem}/{0-17o}
C{word}/{mnemonic}	C{0-999d}/{mnem}/{0-17o}

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): CU, CD, CN, OV, UF

Integer (16-bit) Files

Syntax	Ranges
N{file}:{word}	N{0-999d}:{0-9999d}
N{file}:{word}/{bit}	N{0-999d}:{0-9999d}/{0-17o}
N{file}/{bit}	N{0-999d}/{0-470377o}

Arrays are supported.

Integer (32-bit) Files

Syntax	Ranges
H{file}:{word}	H{0-999d}:{0-9999d}
H{file}:{word}/{bit}	H{0-999d}:{0-9999d}/{0-37o}
H{file}/{bit}	H{0-999d}/{0-1160777o}

Arrays are supported.

Float Files

Syntax	Ranges
F{file}:{word}	F{0-999d}:{0-9999d}

Decimal Files

Syntax	Ranges
D{file}:{word}	D{0-999d}:{0-9999d}
D{file}:{word}/{bit}	D{0-999d}:{0-9999d}/{0-17o}
D{file}/{bit}	D{0-999d}/{0-470377o}

Arrays are supported.

Pointers Files

Syntax	Ranges
P{mnemonic}:{word}	P{mnem}:{0-9999d}

Valid mnemonics

SEC, FIL, WRD, IND

PID Files

Syntax	Ranges
PD{file}:{word}.{mnemonic}/{bit}	PD{3-255d}:{0-255d}.{mnem}/{0-15d}

Valid mnemonics

TM, AM, CM, OL, RG, SC, TF, DA, DB, UL, LL, SPPV, DN, EN, SPS, KC, Ti, TD, MAXS, MINS, ZCD, CVH, CVLLUT, SPV, CVP

Message Files

Syntax	Ranges
MG{file}:{word}.{mnemonic}/{bit}	MG{3-255d}:{0-255d}.{mnem}/{0-15d}

Valid mnemonics

IA, RBL, LBN, RBN, CHN, NOD, MTO, NB, TFT, TFN, ELE, SELTO, CO, EN, RN, EW, ER, DN, ST

Programmable Limit Switch Files

Syntax	Ranges
PLS{file}:{word}.{mnemonic}/{bit}	PLS{3-255d}:{0-255d}.{mnem}/{0-15d}

Valid mnemonics

HIP, LOP, OHD, OLD

PLC-5 Family

This section describes the logical addressing for PLC files supported by the PLC-5 family.

Input Files

Syntax	Ranges
I:{element}	I:{0-277o}
I:{element}/{bit}	I:{0-277o}/{0-17o}
I:{element}/{bit},{cnt}	I:{0-277o}/{0-17o},{1-16d}
I/{bit}	I/{0-5777o}

Arrays are supported.

Output Files

Syntax	Ranges
O:{element}	O:{0-277o}
O:{element}/{bit}	O:{0-277o}/{0-17o}
O:{element}/{bit},{cnt}	O:{0-277o}/{0-17o},{1-16d}
O/{bit}	O/{0-5777o}

Arrays are supported.

Status Files

Syntax	Ranges
S:{element}	S:{0-128d}
S:{element}/{bit}	S:{0-128d}/{0-15d}
S:{element}/{bit},{cnt}	S:{0-128d}/{0-15d},{1-16d}
S/{bit}	S/{0-2063d}

Arrays are supported.

Timer Files

Syntax	Ranges
T{file}:{element}/{bit}	T{3-999d}:{0-999d}/{0-15d}
T{file}:{element}/{bit},{cnt}	T{3-999d}:{0-999d}/{0-15d},{1-16d}
T{file}:{element}.{mnem}/{bit}	T{3-999d}:{0-999d}.{mnem}/{0-15d}
T{file}: {element}.{mnem}/{bit},{cnt}	T{3-999d}: {0-999d}.{mnem}/{0-15d},{1-16d}
T{file}:{element}/{mnem}/{bit}	T{3-999d}:{0-999d}/{mnem}/{0-15d}
T{file}: {element}/{mnem}/{bit},{cnt}	T{3-999d}: {0-999d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): DN, TT, EN

Counter Files

Syntax	Ranges
C{file}:{element}/{bit}	C{3-999d}:{0-999d}/{0-15d}
C{file}:{element}/{bit},{cnt}	C{3-999d}:{0-999d}/{0-15d},{1-16d}
C{file}:{element}.{mnem}/{bit}	C{3-999d}:{0-999d}.{mnem}/{0-15d}
C{file}: {element}.{mnem}/{bit},{cnt}	C{3-999d}: {0-999d}.{mnem}/{0-15d},{1-16d}
C{file}:{element}/{mnem}/{bit}	C{3-999d}:{0-999d}/{mnem}/{0-15d}
C{file}: {element}/{mnem}/{bit},{cnt}	C{3-999d}: {0-999d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): UN, OV, DN, CD, CU, (UA)

Control Files

Syntax	Ranges
R{file}:{element}/{bit}	R{3-999d}:{0-999d}/{0-15d}
R{file}:{element}/{bit},{cnt}	R{3-999d}:{0-999d}/{0-15d},{1-16d}
R{file}:{element}.{mnem}/{bit}	R{3-999d}:{0-999d}.{mnem}/{0-15d}
R{file}: {element}.{mnem}/{bit},{cnt}	R{3-999d}: {0-999d}.{mnem}/{0-15d},{1-16d}
R{file}:{element}/{mnem}/{bit}	R{3-999d}:{0-999d}/{mnem}/{0-15d}
R{file}: {element}/{mnem}/{bit},{cnt}	R{3-999d}: {0-999d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): LEN, POS

Boolean fields (read only): FD, IN, UL, ER, EM, DN, EU, EN

Integer Files

Syntax	Ranges
N{file}:{element}	N{3-999d}:{0-999d}
N{file}:{element}/{bit}	N{3-999d}:{0-999d}/{0-15d}
N{file]:{element}/{bit},{cnt}	N{3-999d}:{0-999d}/{0-15d},{1-16d}
N{file}/{bit}	N{3-999d}/{0-15999d}

Arrays are supported.

Binary Files

Syntax	Ranges
B{file}:{element}	B{3-999d}:{0-999d}
B{file}:{element}/{bit}	B{3-999d}:{0-999d}/{0-15d}
B{file}:{element}/{bit},{cnt}	B{3-999d}:{0-999d}/{0-15d},{1-16d}
B[file}/{bit}	B{3-999d}/{0-15999d}

Arrays are supported.

Float Files

Syntax	Ranges
F{file}:{element}	N{3-999d}:{0-999d}

Arrays are supported.

ASCII Files

Syntax	Ranges
A{file}:{element}	A{3-999d}:{0-999d}
A{file}:{element}/{bit}	A{3-999d}:{0-999d}/{0-15d}
A{file}:{element}/{bit},{cnt}	A{3-999d}:{0-999d}/{0-15d},{1-16d}
A{file}/{bit}	A{0-999d}/{0-15999d}

Arrays are supported.

ASCII String Files

Syntax	Ranges
ST{file}:{element}	ST{3-999d}:{0-779d}
ST{file}:{element},{cnt}	ST{3-999d}:{0-779d},{1-16d}
ST{file}:{element}.{mnem}/{bit}	ST{3-999d}:{0-779d}.{mnem}/{0-15d}
ST{file}: {element}.{mnem}/{bit},{cnt}	ST{3-999d}: {0-779d}.{mnem}/{0-15d},{1-16d}
ST{file}:{element}/{mnem}/{bit}	ST{3-999d}:{0-779d}/{mnem}/{0-15d}
ST{file}: {element}/{mnem}/{bit},{cnt}	ST{3-999d}: {0-779d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): LEN

Decimal Files

Syntax	Ranges
D{file}:{element}	D{3-999d}:{0-999d}
D{file}:{element}/{bit}	D{3-999d}:{0-999d}/{0-15d}
D{file}:{element}/{bit},{cnt}	D{3-999d}:{0-999d}/{0-15d},{1-16d}
D{file}/{bit}	D{0-999d}/{0-15999d}

Arrays are supported.

PID Files

Syntax	Ranges
PD{file}:{element}/{bit}	PD{3-999d}:{0-398}/{0-15d}
PD{file}:{element}/{bit},{cnt}	PD{3-999d}:{0-398}/{0-15d},{1-32d}
PD{file}:{element}.{mnem}/{bit}	PD{3-999d}:{0-398d}.{mnem}/{0-15d}
PD{file}: {element}.{mnem}/{bit},{cnt}	PD{3-999d}: {0-398d}.{mnem}/{0-15d},{1-32d}
PD{file}:{element}/{mnem}/{bit}	PD{3-999d}:{0-398d}/{mnem}/{0-15d}
PD{file}: {element}/{mnem}/{bit},{cnt}	PD{3-999d}: {0-398d}/{mnem}/{0-15d},{1-32d}

Valid mnemonics

DATA[0]-DATA[13], ADDR[0]-ADDR[3]

Float fields (read/write): SP, KP, KI, KD, BIAS, MAXS, MINS, MAXO, MINO, UPD, PV, OUTPVH, PVL, DVP, DVN, PVDB, MAXI, MINI, TIE, DB, SO, ERR, DVDB

Boolean fields (read\write): EN, CT, CL, PVT, DO, SWM, CA, MO, PE, INI, SPOR, OLHEWD, DVNA, PVLA, PVHA, OLL, DVPA

Message Files

Syntax	Ranges
MG{file}:{element}/{bit}	MG{3-999d}:{0-584d}/{0-15d}
MG{file}:{element}/{bit},{cnt}	MG{3-999d}:{0-584d}/{0-15d},{1-16d}
MG{file}:{element}.{mnem}/{bit}	MG{3-999d}:{0-584d}.{mnem}/{0-15d}
MG{file}: {element}.{mnem}/{bit},{cnt}	MG{3-999d}: {0-584d}.{mnem}/{0-15d},{1-16d}
MG{file}:{element}/{mnem}/{bit}	MG{3-999d}:{0-584d}/{mnem}/{0-15d}
MG{file}: {element}/{mnem}/{bit},{cnt}	MG{3-999d}: {0-584d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

DATA[0]-DATA[51]

Word fields (read/write): ERR, RLEN, DLEN

Boolean fields (read\write): EN, ST, DN, ER, CO, EW, NR, TO, AD, AE

Block Transfer Files

Syntax	Ranges
BT{file}:{element}/{bit}	BT{3-999d}:{0-584d}/{0-15d}
BT{file}:{element}/{bit},{cnt}	BT{3-999d}:{0-584d}/{0-15d},{1-16d}
BT{file}:{element}.{mnem}/{bit}	BT{3-999d}:{0-584d}.{mnem}/{0-15d}
BT{file}: {element}.{mnem}/{bit},{cnt}	BT{3-999d}: {0-584d}.{mnem}/{0-15d},{1-16d}
BT{file}:{element}/{mnem}/{bit}	BT{3-999d}:{0-584d}/{mnem}/{0-15d}
BT{file}: {element}/{mnem}/{bit},{cnt}	BT{3-999d}: {0-584d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): FILE, RLEN, DLEN, ELEM, RGS, EC, IDX, PLEN, TOUT

Boolean fields (read\write): RW, ST, DN, ER, CO, EW, TO, RW, EN, NR, AD, AE

SFC Status Files

Syntax	Ranges
SC{file}:{element}/{bit}	SC{3-999d}:{0-398d}/{0-15d}
SC{file}:{element}/{bit},{cnt}	SC{3-999d}:{0-398d}/{0-15d},{1-16d}
SC{file}:{element}.{mnem}/{bit}	SC{3-999d}:{0-398d}.{mnem}/{0-15d}
SC{file}: {element}.{mnem}/{bit},{cnt}	SC{3-999d}: {0-398d}.{mnem}/{0-15d},{1-16d}
SC{file}:{element}/{mnem}/{bit}	SC{3-999d}:{0-398d}/{mnem}/{0-15d}
SC{file}: {element}/{mnem}/{bit},{cnt}	SC{3-999d}: {0-398d}/{mnem}/{0-15d},{1-16d}

Valid mnemonics

Word fields (read/write): PRE, TIM

Boolean fields (read/write): SA, FS, LS, OV, ER, DN

PLC-5/10, 5/12, 5/15, 5/25

These PLC-5s are like the PLC-5 family with the following exceptions:

1. Block Transfer, Message, PID, SFC Status and ASCII String files are not available.
2. The maximum word number for a Status File is 31.

SLC 5/04, 5/05

Input Files

Syntax	Ranges
I:{slot}.{word}/{bit}	I:{0-30d}.{0-255d}/{0-15d}
I:{slot}.{word}	I:{0-30d}.{0-255d}
I:{slot}/{bit}	I:{0-30d}/{0-4095d}

Output Files

Syntax	Ranges
O:{slot}.{word}/{bit}	O:{0-30d}.{0-255d}/{0-15d}
O:{slot}.{word}	O:{0-30d}.{0-255d}
O:{slot}/{bit}	O:{0-30d}/{0-4095d}

Decimal Files

Syntax	Ranges
D{file]:{word}	D{3-255d}:{0-255d}
D{file}:{word}/{bit}	D{3-255d}:{0-255d}/{0-15d}
D{file}/{bit}	D{3-255d}/{0-4095d}

Arrays are supported.

Status Files

Syntax	Ranges
S:{word}	S:{0-83d}
S:{word}/{bit}	S:{0-83d}/{0-15d}
S/{bit}	S/{0-1343d}

Arrays are supported.

Integer Files

Syntax	Ranges
N{file}:{word}	N{3-255d}:{0-255d}
N{file}:{word}/{bit}	N{3-255d}:{0-255d}/{0-15d}
N{file}/{bit}	N{3-255d}/{0-4095d}

Arrays are supported.

ASCII Files

Syntax	Ranges
A{file}:{word}	A{3-255d}:{0-255d}
A{file}:{word}/{bit}	A{3-255d}:{0-255d}/{0-15d}
A{file}/{bit}	A{3-255d}/{0-4095d}

Arrays are supported.

Binary Files

Syntax	Ranges
B{file}:{word}	B{3-255d}:{0-255d}
B{file}:{word}/{bit}	B{3-255d}:{0-255d}/{0-15d}
B{file}/{bit}	B{3-255d}/{0-4095d}

Arrays are supported.

ASCII String Files

Syntax	Ranges
ST{file}:{word}.{mnem}/{bit}	ST{3-255d}:{0-255d}.{mnem}/{0-15d}
ST{file}:{word}/{mnem}/{bit}	ST{3-255d}:{0-255d}/{mnem}/{0-15d}

Valid mnemonics

Word fields (read/write): LEN

Float Files

Syntax	Ranges
F{file]:{word}	F{3-255d}:{0-255d}

Timer Files

Syntax	Ranges
T{file}:{structure}.{mnem}/{bit}	T{3-255d}:{0-255d}.{mnem}/{0-15d}
T{file}:{structure}/{mnem}/{bit}	T{3-255d}:{0-255d}/{mnem}/{0-15d}

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): DN, TT, EN

Counter Files

Syntax	Ranges
C{file}:{structure}.{mnem}/{bit}	C{3-255d}:{0-255d}.{mnem}/{0-15d}
C{file}:{structure}/{mnem}/{bit}	C{3-255d}:{0-255d}/{mnem}/{0-15d}

Valid mnemonics

Word fields (read/write): ACC, PRE

Boolean fields (read only): UN, OV, DN, CD, CU

Control Files

Syntax	Ranges
R{file}:{structure}.{mnem}/{bit}	R{3-255d}:{0-255d}.{mnem}/{0-15d}
R{file}:{structure}/{mnem}/{bit}	R{3-255d}:{0-255d}/{mnem}/{0-15d}

Valid mnemonics

Word fields (read/write): LEN, POS

Boolean fields (read only): FD, IN, UL, ER, EM, DN, EU, EN

SLC-500 Fixed I/O, SLC 5/01, 5/02, 5/03

These SLCs are like the SLC5/04, 5/05 family with the following exceptions:

1. Floats and ASCII String files are not available.
2. SLC 500 and 5/01 support only status words 0-15. SLC 5/02 support only words 0-32.
3. The SLC 500 supports the UA mnemonic for counters.

MicroLogix

The MicroLogix Processors are similar to the SLCs with the following exceptions:

1. MicroLogix 1200 and 1500 add Message files, PID Files and Programmable Limit Switch files

2. The MicroLogix 1000 has a fixed file structure. You cannot add new files and the size of the files are limited.

APPENDIX B: DATA ITEM DUPLICATION WIZARD

To speed up the creation of similarly configured data items, you can easily create multiple data items by duplicating an existing one. To do this, right-click on an existing data item and select ***Duplicate...*** from the context menu. The Data Item Duplication Wizard will open.

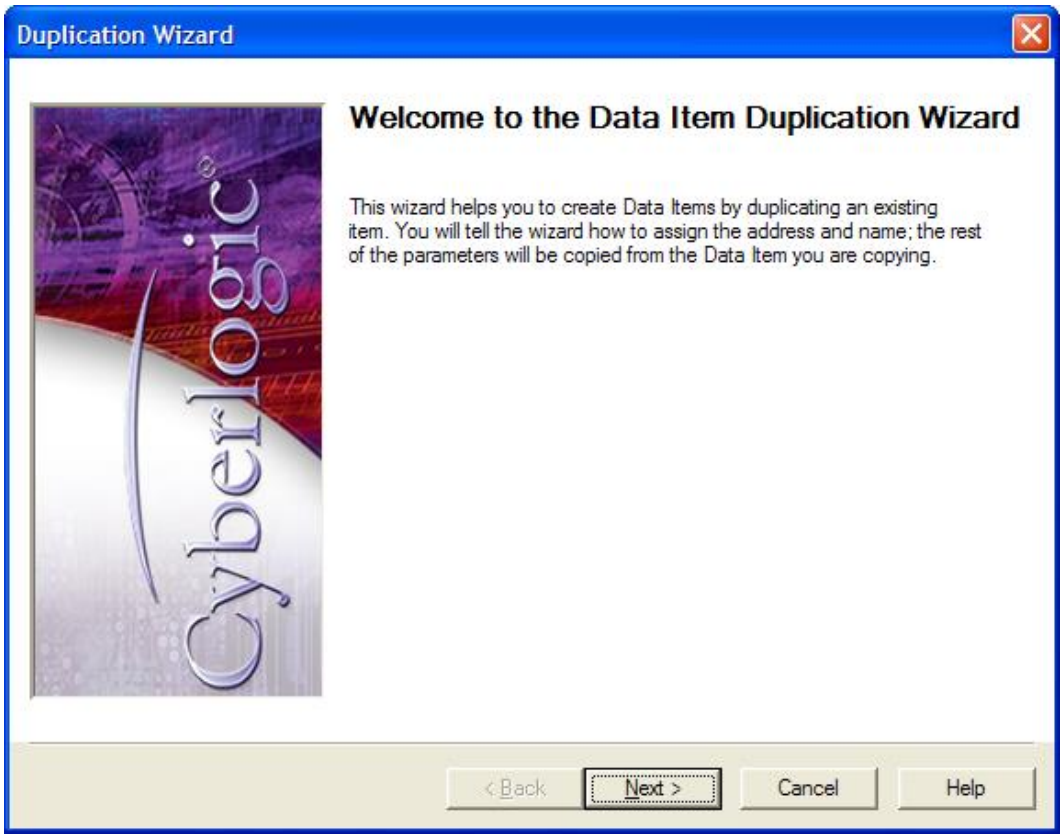
We will present two examples here.

Example 1: Simple Addressing shows how to duplicate input bits. This same procedure would be used for other data using simple addressing.

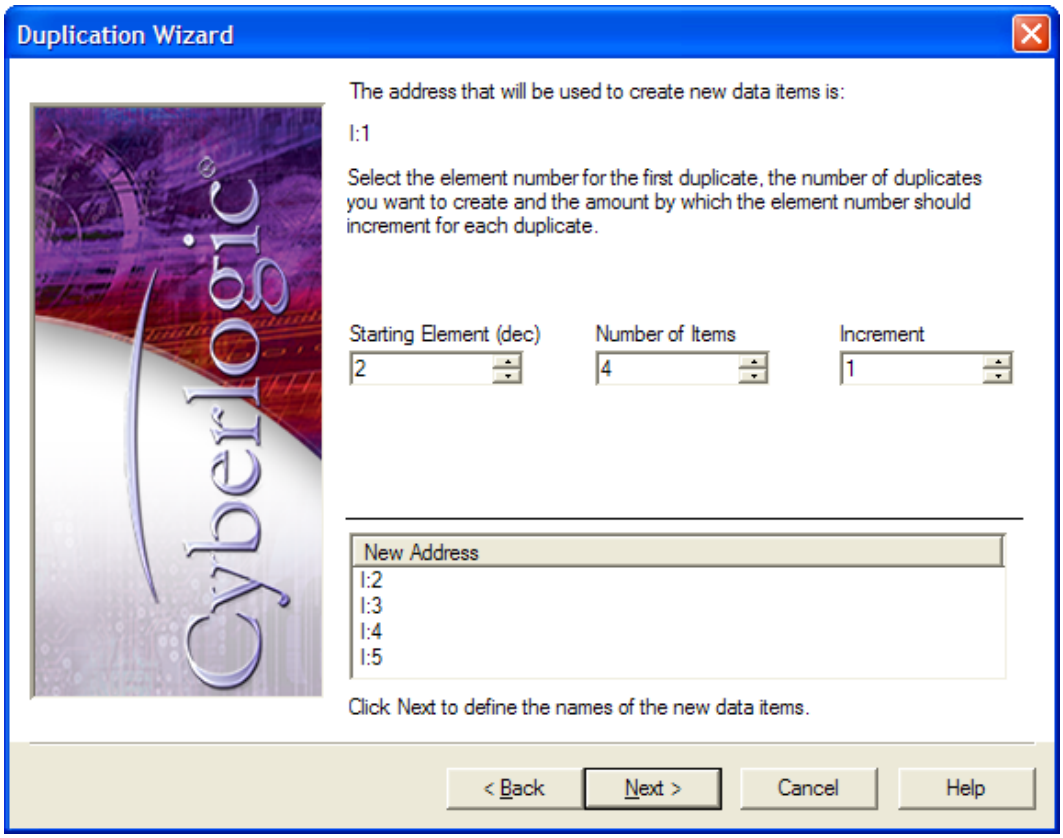
Example 2: Addressing Bits Within Registers is more complex and shows how to handle data addressed as bits within a register or array.

Example 1: Simple Addressing

For this example, we selected for duplication a data item called I_001 that is associated with register I:1 in a PLC.

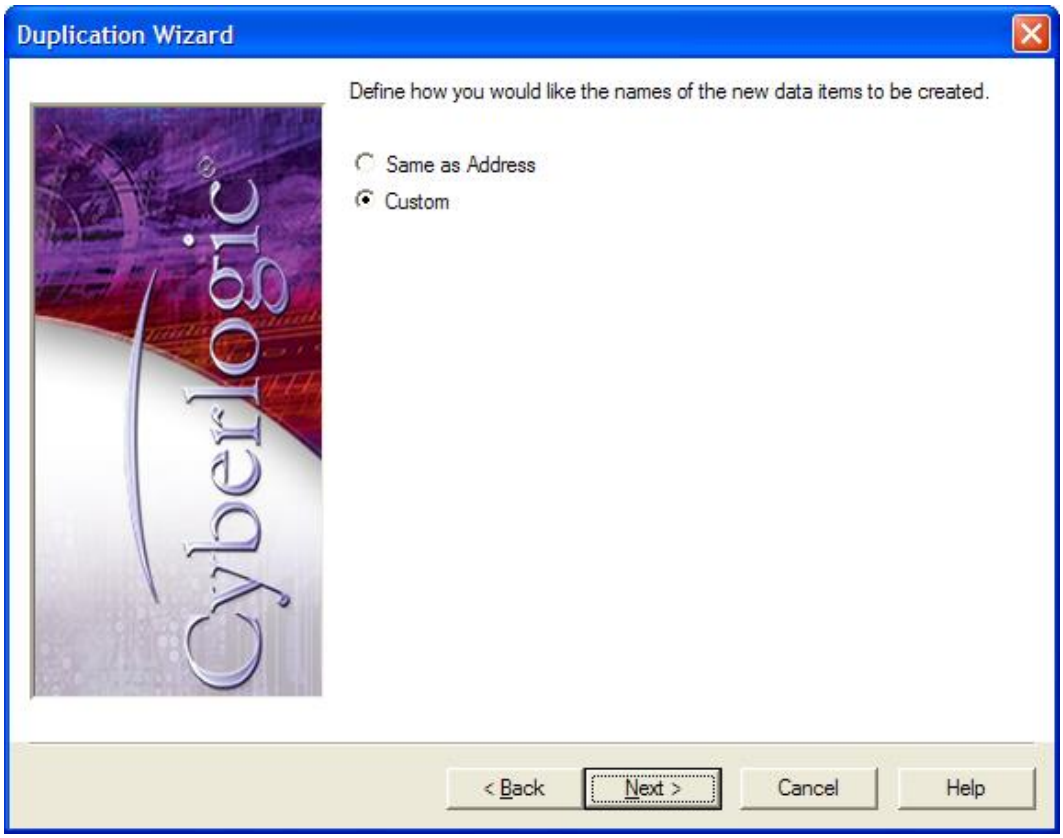


1. On the Welcome screen, click the ***Next*** button to continue.



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.

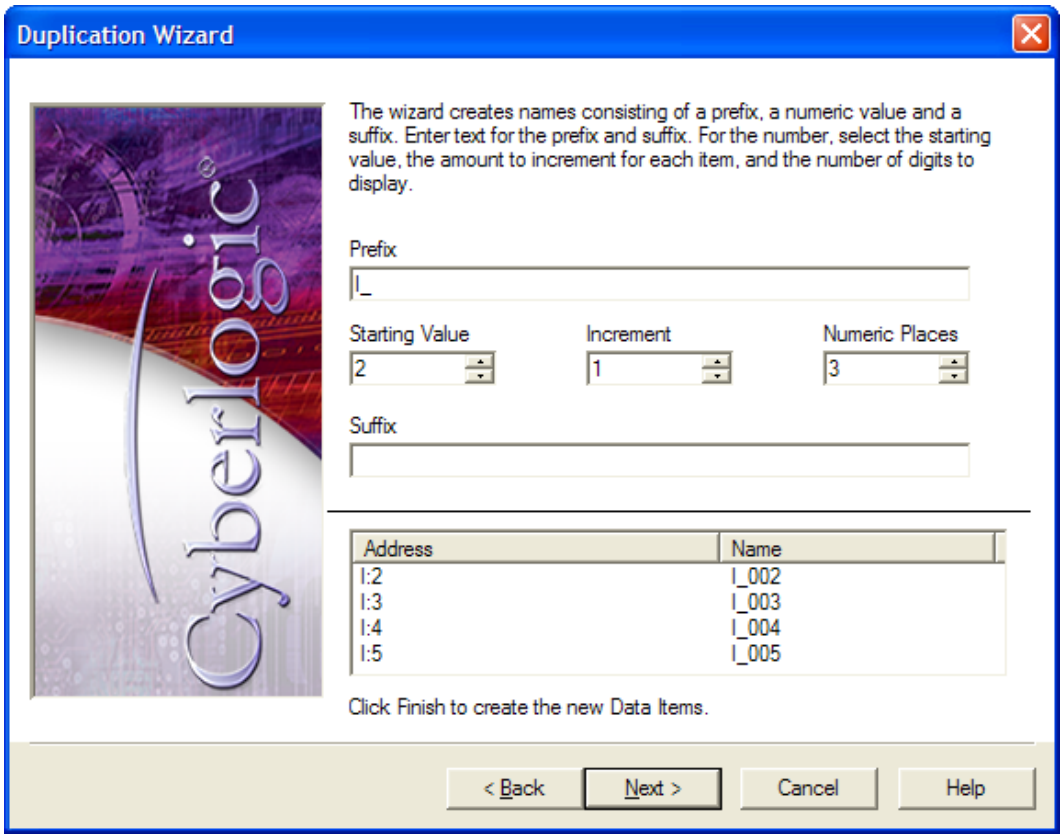
2. Enter ***2*** for the ***Starting Element***.
3. Enter ***4*** for the ***Number of Items***.
4. Enter ***1*** for the ***Increment***.
5. The ***New Address*** box shows you the data items that the wizard will create. Verify that these are correct.
6. Click ***Next*** to continue.



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.

7. Select ***Custom***.

8. Click ***Next*** to continue.



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.

9. Enter ***I_*** in the ***Prefix*** field, to have all of the names start with I_.

   The next three fields define the numeric values to be used.

10. Enter ***2*** as the ***Starting Value***.

11. Enter ***1*** for the ***Increment***.

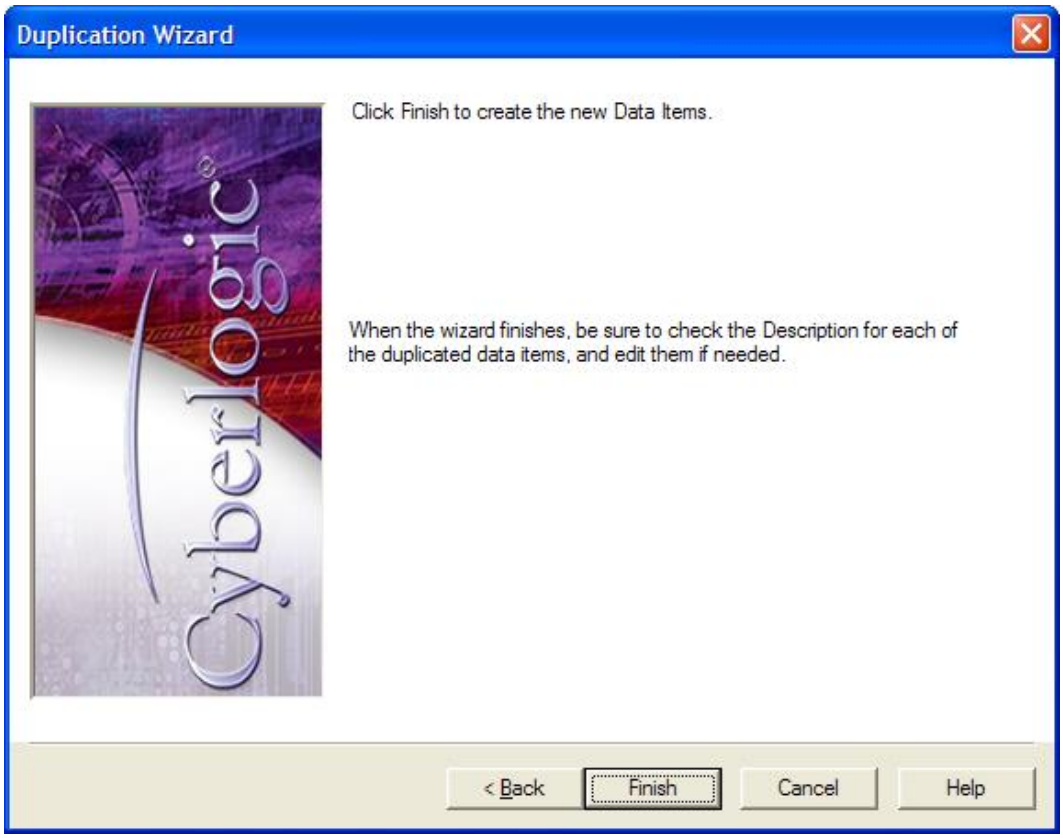
12. Enter ***3*** for ***Numeric Places***.

    These selections cause the duplicates to be numbered consecutively, beginning with 2. It also forces the names to use three digits, inserting leading zeros as needed.

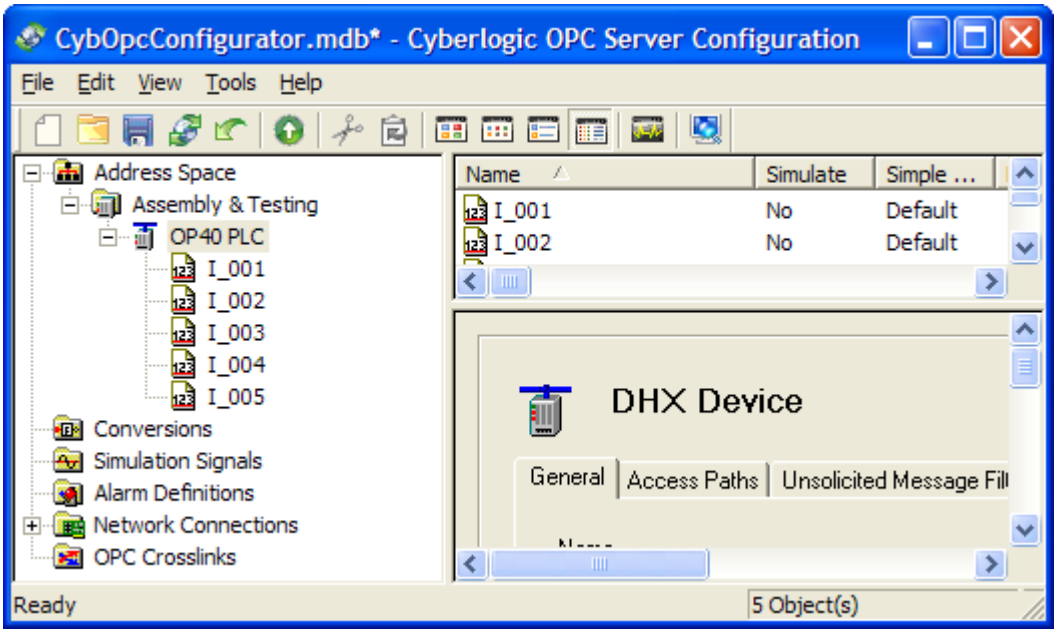
13. Leave the ***Suffix*** field blank, because no suffix is necessary for this naming scheme.

14. The lower window will show you the names that will be used for each data item. Verify that these are correct.

15. Click ***Next*** to continue.



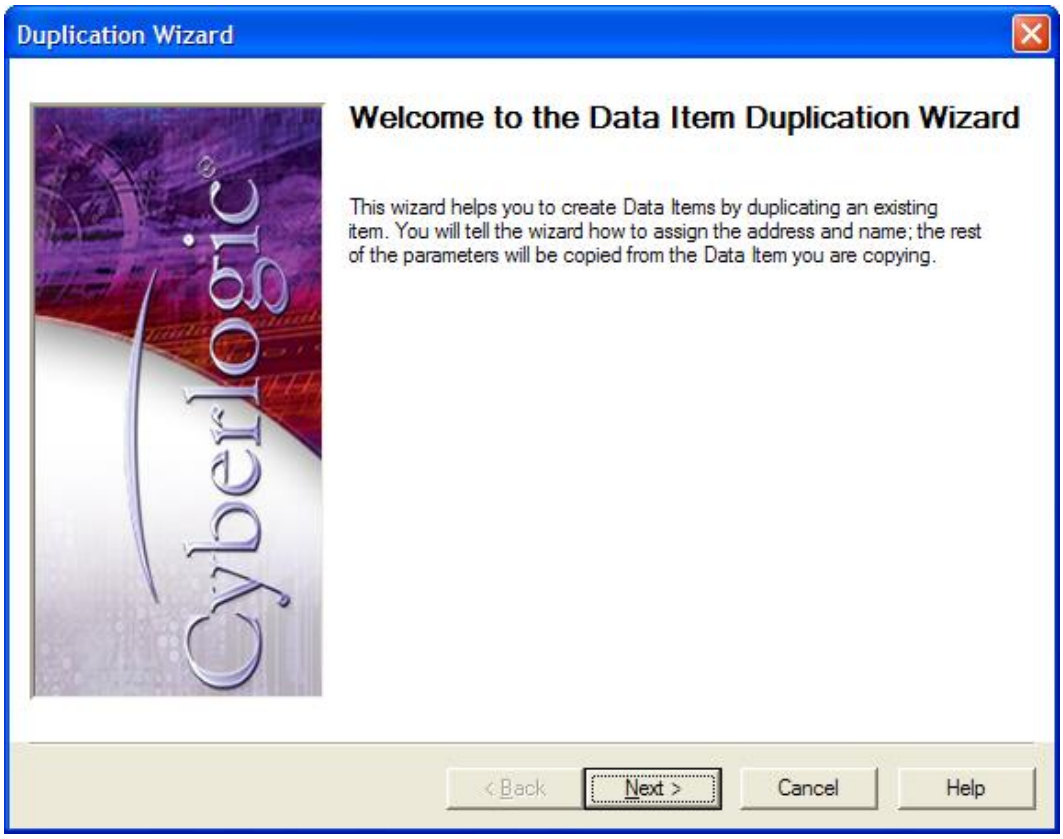
16. Click ***Finish*** to create the duplicate data items and exit the wizard.



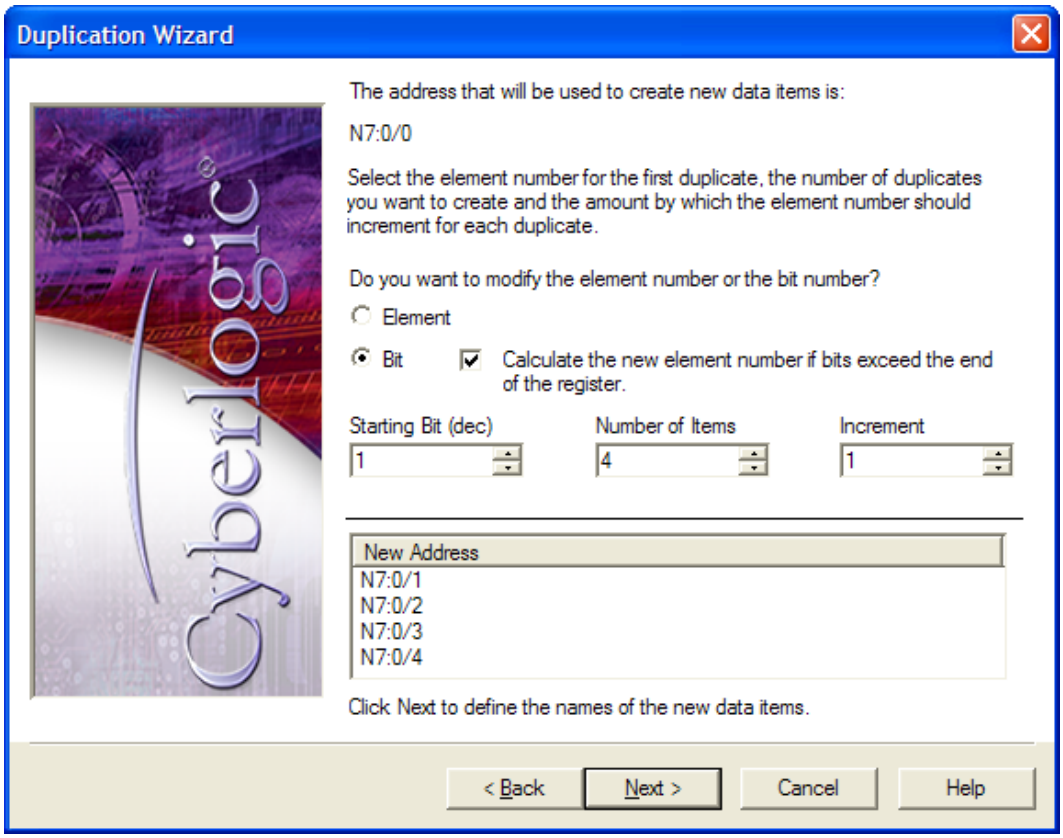
Your duplicates will be created. You will find, however, that the Description field for each duplicate is the same as the original. You may wish to edit these descriptions. You should also check the other parameters to verify that they are set properly.

Example 2: Addressing Bits Within Registers

For this example, we selected a data item called StatusItem0. It is a bit within a word and is addressed as N7:0/0. We want to address additional bits within this same word.



1. On the Welcome screen, click the ***Next*** button to continue.



2. Select ***Bit*** to indicate that you want to address additional bits within the same register, rather than additional registers.

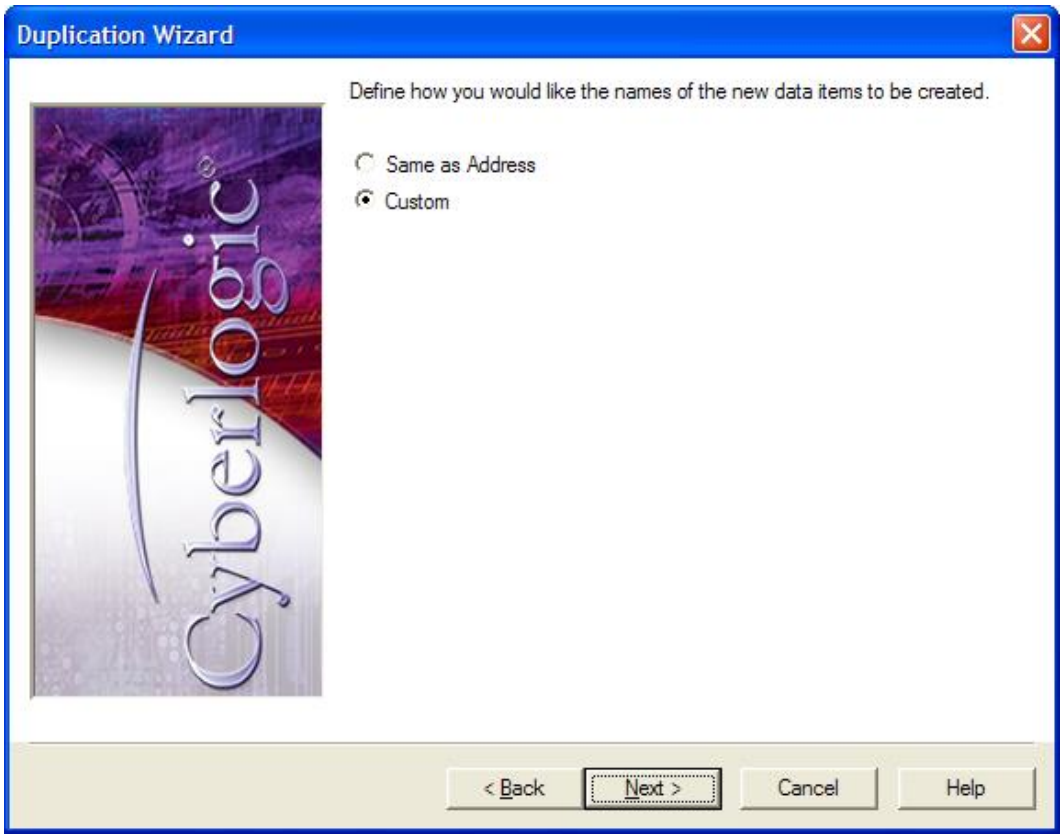
3. Enter ***1*** for the ***Starting Bit*** value, to specify that the first duplicate should be N7:0/1.

4. Enter ***4*** for the ***Number of Items***, to create four duplicates.

5. Enter ***1*** for the ***Increment***, so that you will access consecutive bits.

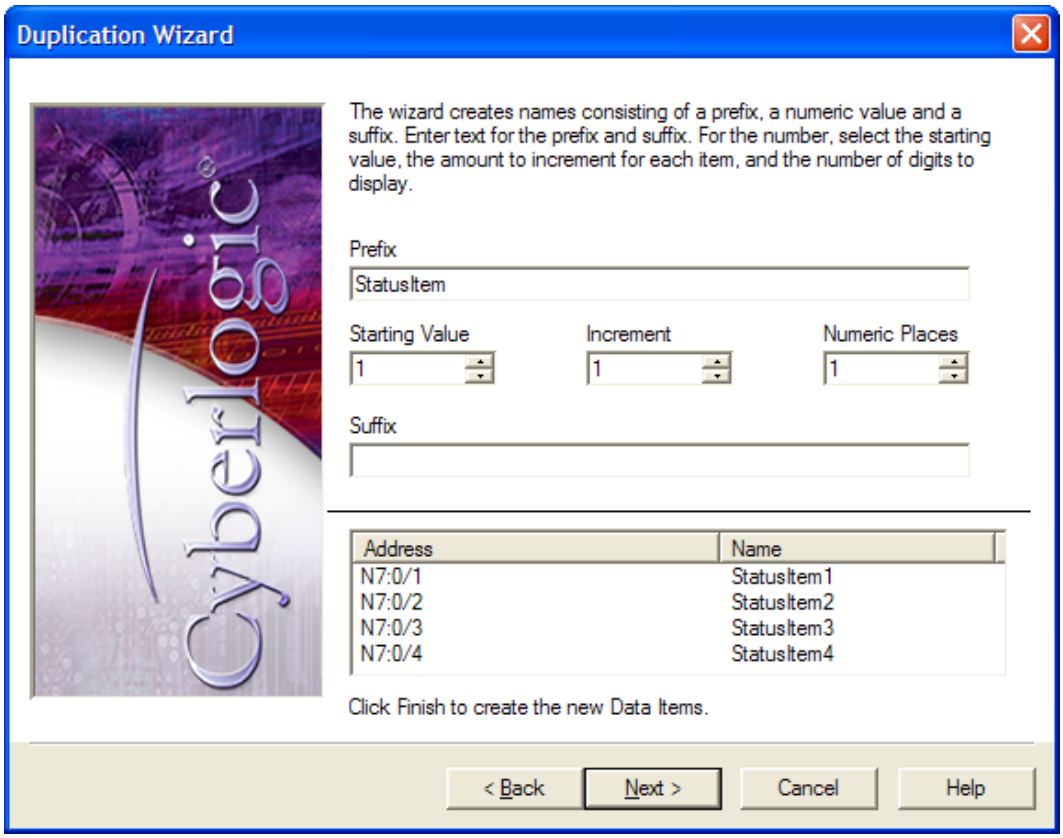
6. The lower window will show the new addresses that will be accessed. Verify that these are correct.

7. Click ***Next*** to continue.



8. Select ***Custom*** to indicate that you want to specify names for the new data items, rather than simply using each item's address as its name.

9. Click ***Next*** to continue.



The name for each item will be built by concatenating a prefix, a number and a suffix.

10. Enter ***StatusItem*** as the ***Prefix***.

    The next three fields specify the number.

11. Enter ***1*** as the ***Starting Value***.

12. Enter ***1*** as the ***Increment***.

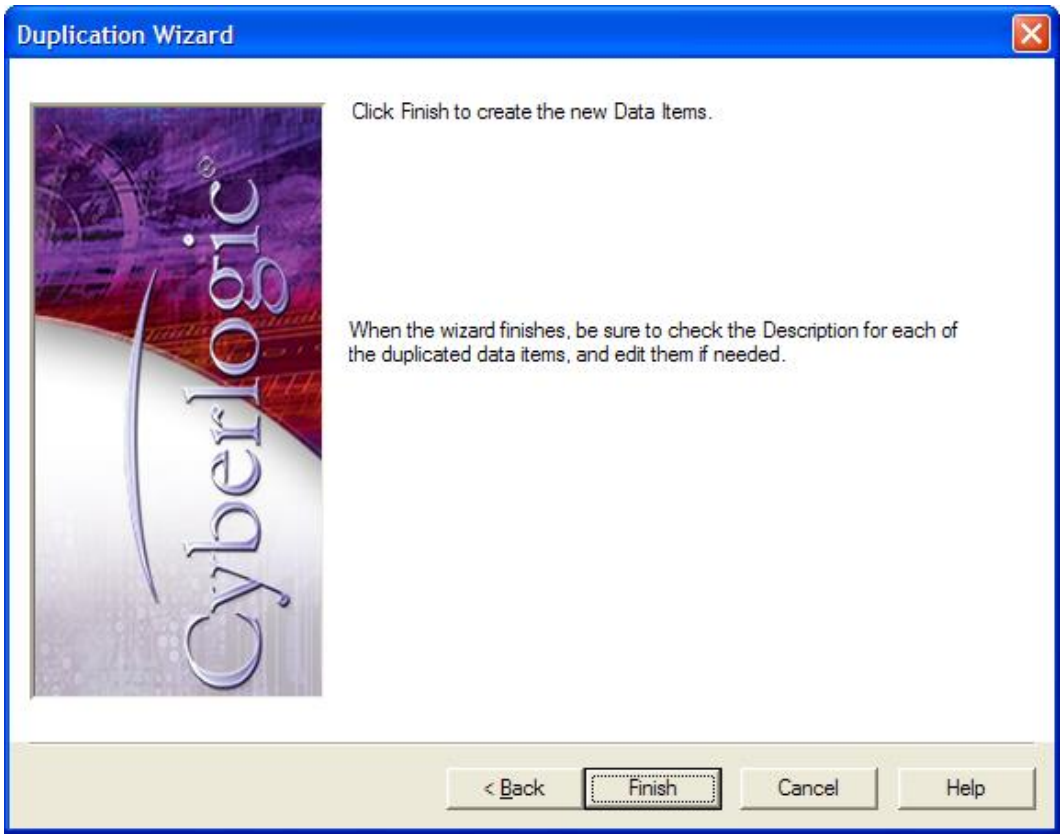
    The data items will therefore be numbered consecutively between 1 and 4.

13. Select ***1*** for the ***Numeric Places,*** because there is no need to force the use of leading zeros in the name.

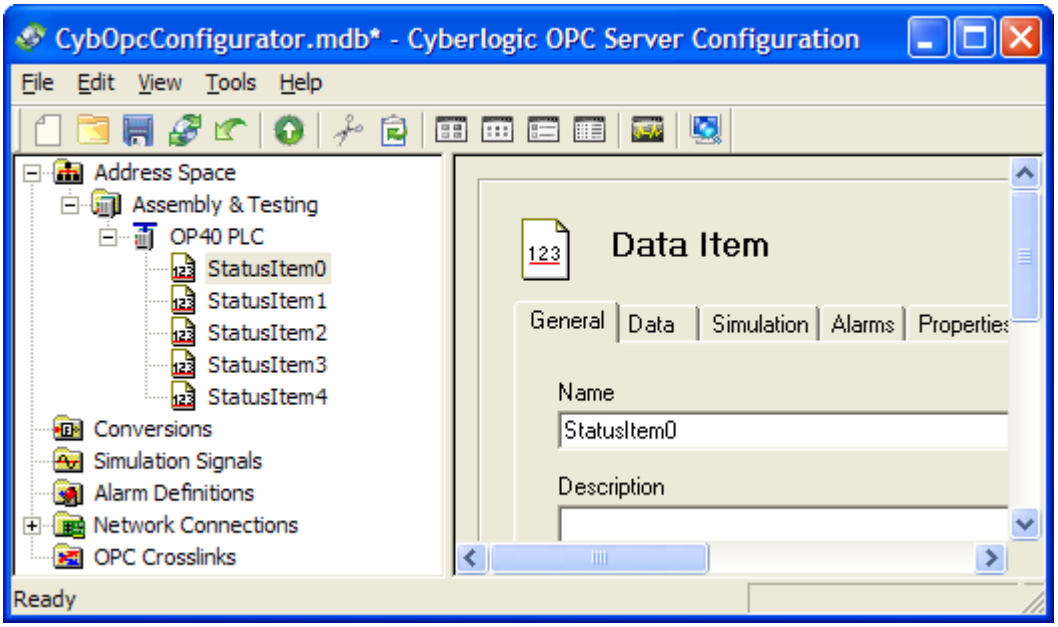
14. The ***Suffix*** field is optional and is not used in this example, so leave it blank.

15. The wizard will show you the names that will be assigned to each address. Verify that these are correct.

16. Click ***Next*** to continue.



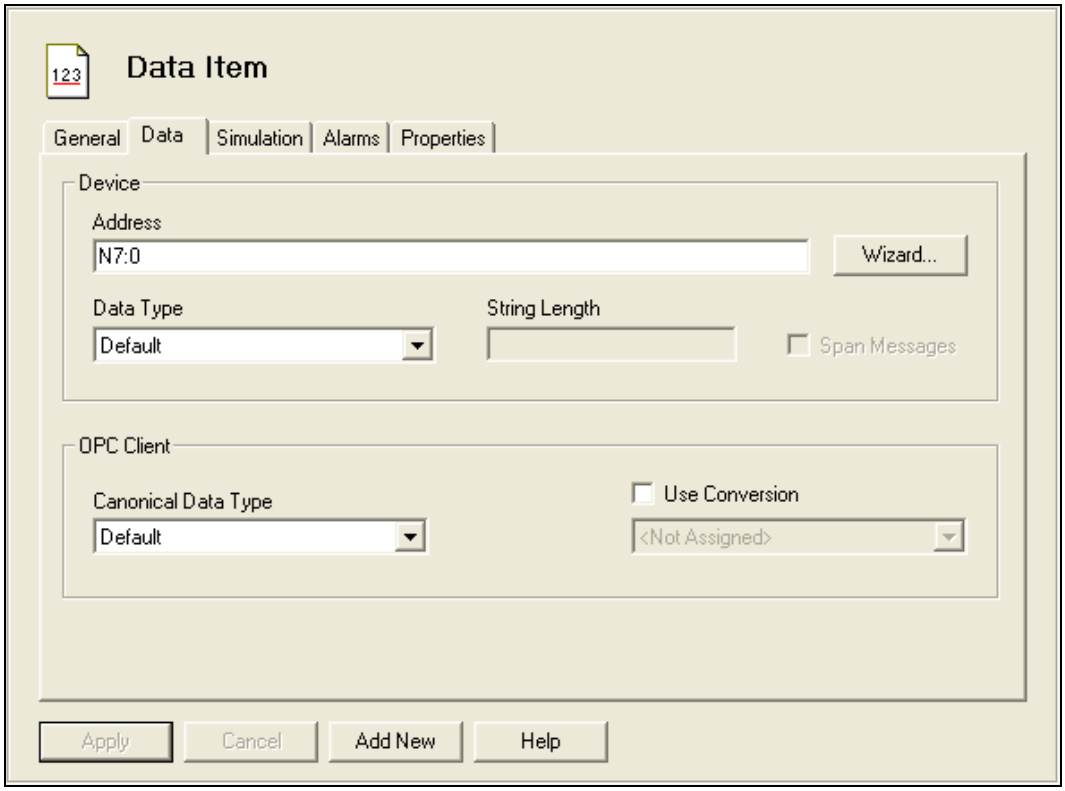
17. Click ***Finish*** to create the duplicate data items and exit the wizard.



Your duplicates will be created. You will find, however, that the Description field for each duplicate is the same as the original. You may wish to edit these descriptions. You should also check the other parameters to verify that they are set properly.

APPENDIX C: ADDRESS WIZARD

The Address Wizard will help you to define the correct address for the data you want to access. To activate the Address Wizard, click the ***Wizard...*** button on the Data tab of the data item dialog.



This appendix provides three examples of the use of the Address Wizard.

Example 1: Simple Addressing covers setting up a simple address for a single register.

Example 2: Accessing Data Within a Structure shows how to access a part of a data structure, in this case, the accumulated value of a timer.

Example 3: Arrays of Registers shows how to address registers as an array.

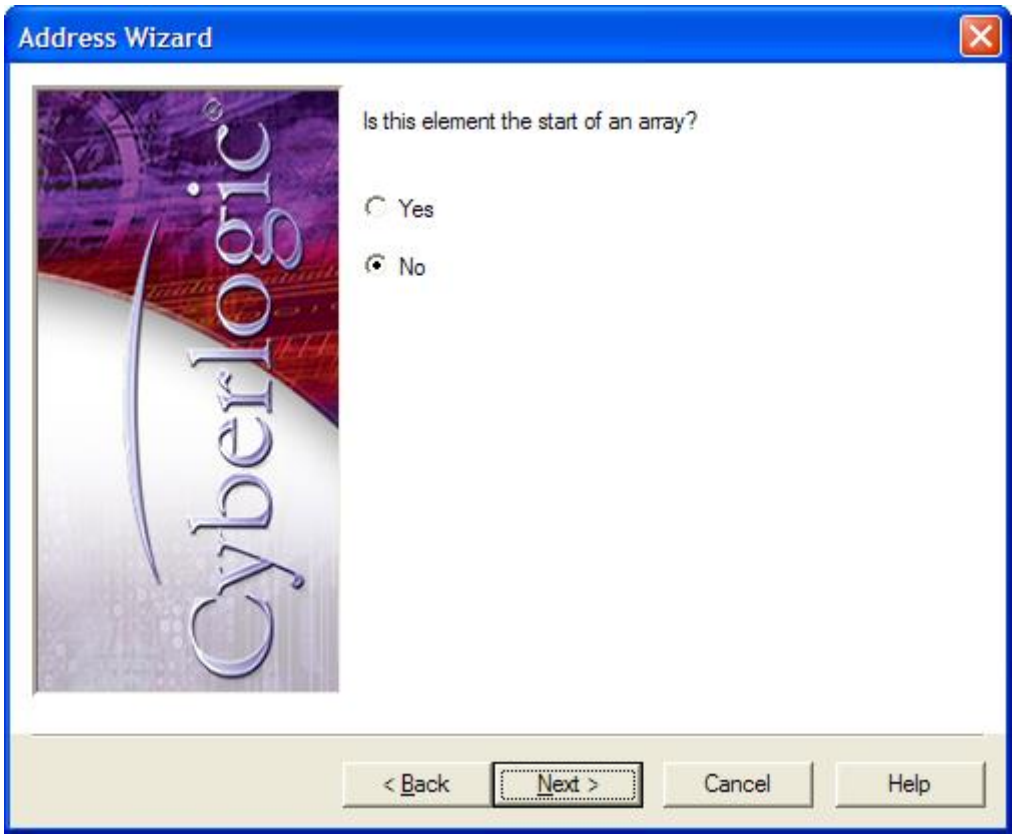
Example 1: Simple Addressing

This example shows how to set up the addressing for a single register.

1. From the Welcome screen, click ***Next*** to continue.

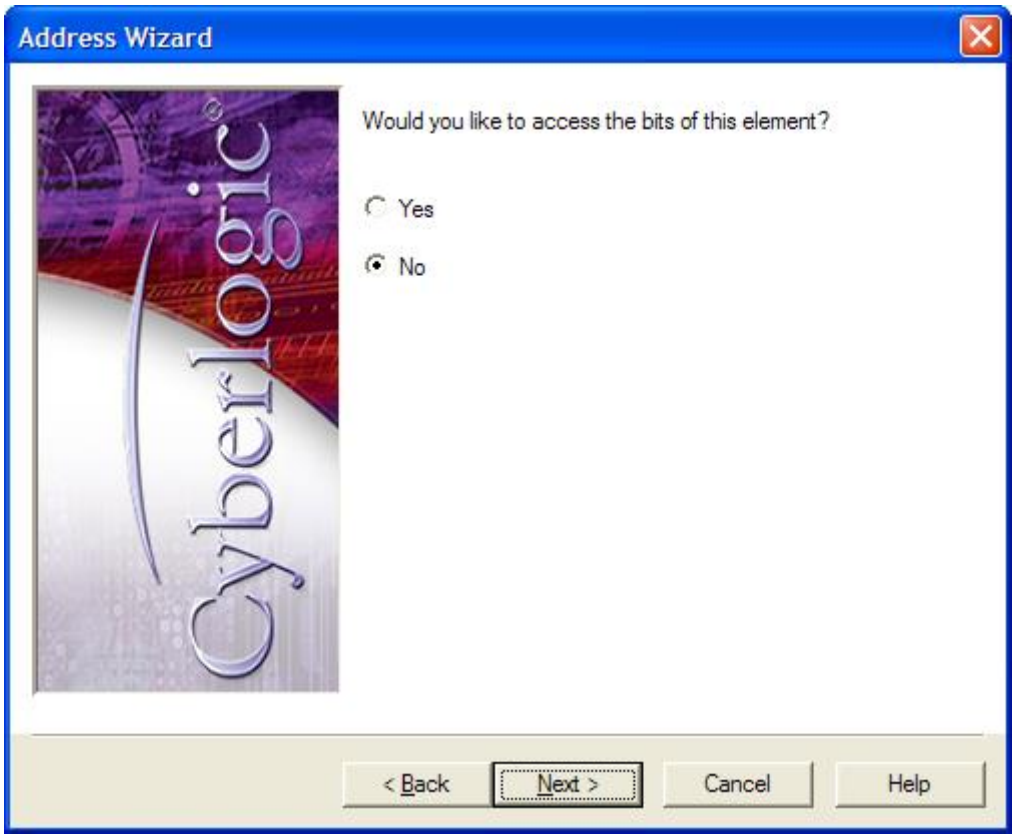
On this screen, you must select the specific register you want to access. The choices will vary depending upon the type of PLC you are addressing.

2. For this example, select ***Integer*** as the file type.
3. Select ***7*** for the file number.
4. Enter ***0*** as the element number. This gives us register 0 of integer file 7.
5. Click ***Next***.



For this example, we do not want to access the data as an array.

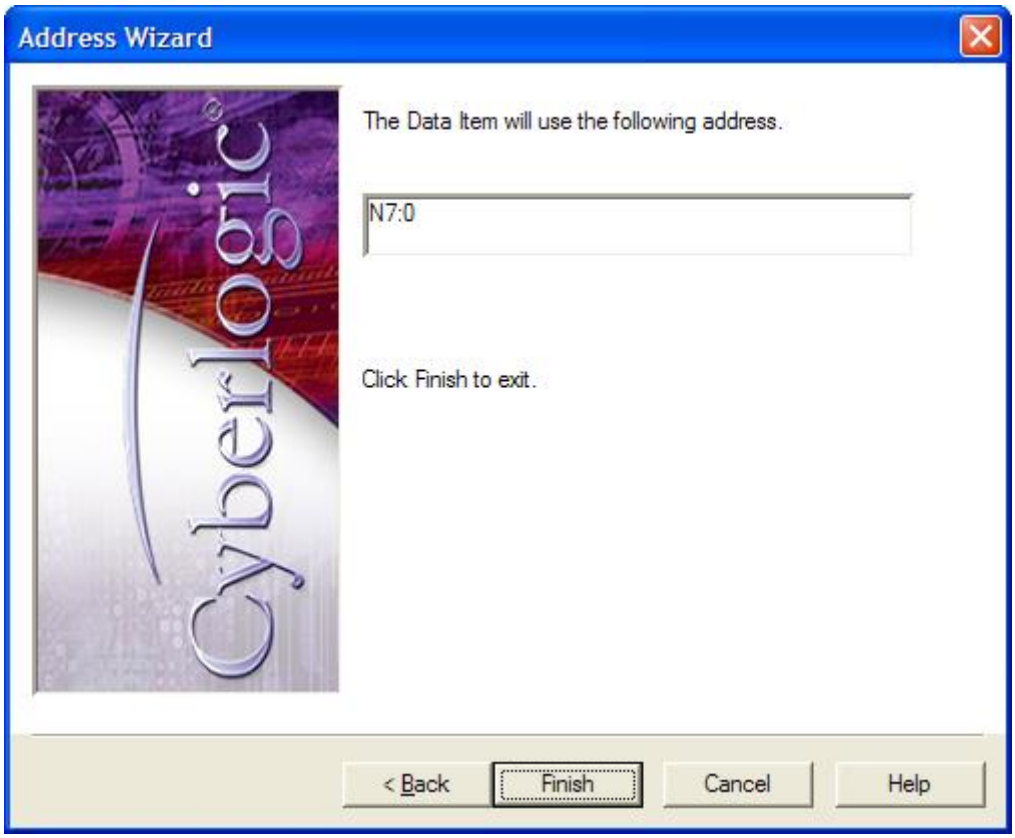
6. Select ***No***.
7. Click ***Next***.



It is possible to access individual bits or groups of bits within the register, but we want to access the register as a word.

8. Select ***No***.

9. Click ***Next***.



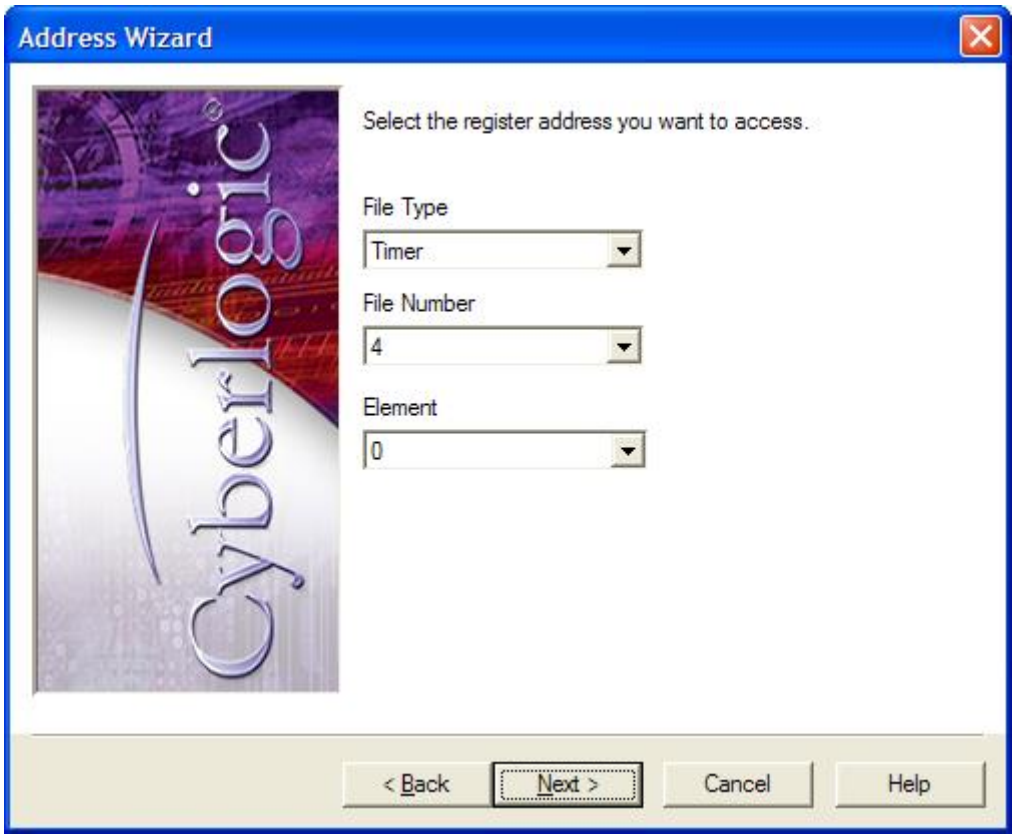
This screen shows the result of your selections. The register address to be created is N7:0.

10. Click ***Finish***.

Example 2: Accessing Data Within a Structure

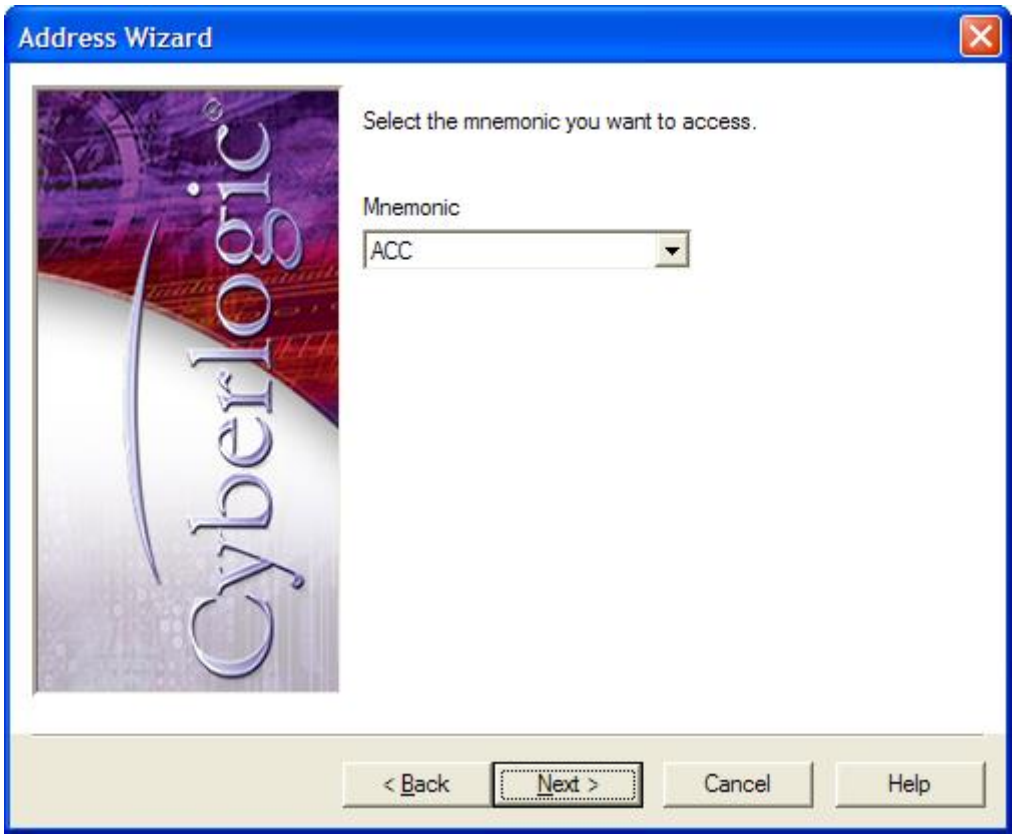
This example shows how to set up the addressing for a value within a data structure. We will access the accumulated value of a timer.

1. From the Welcome screen, click ***Next*** to continue.



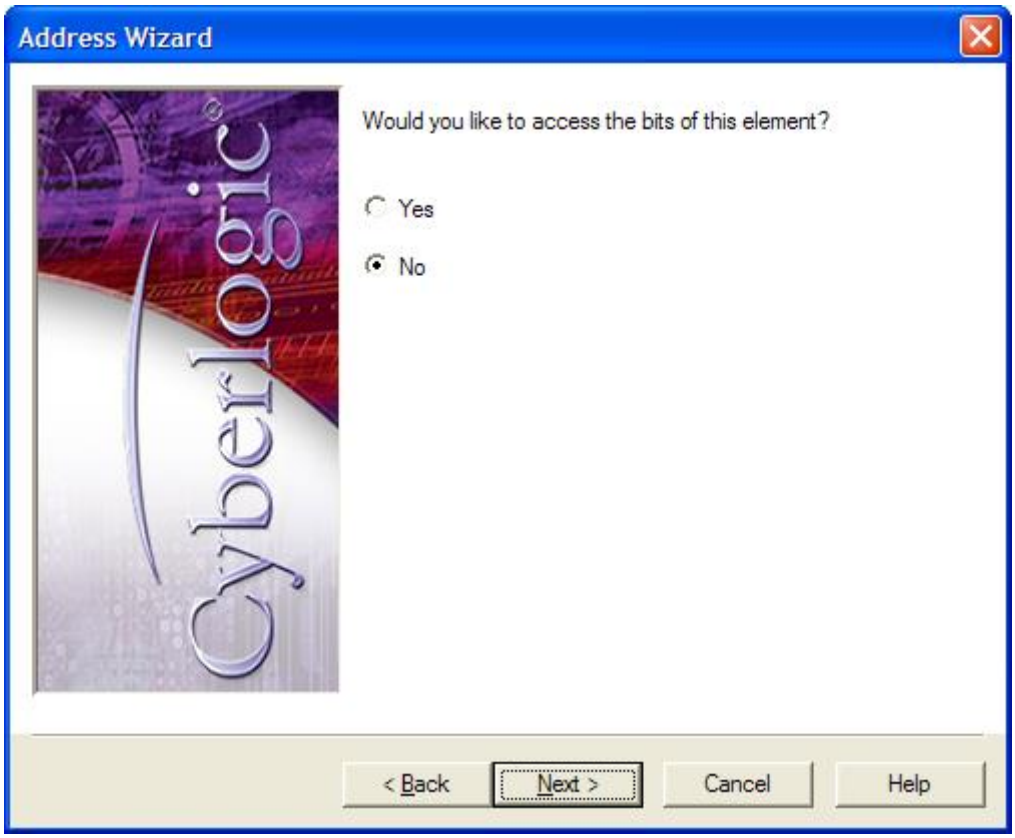
On this screen, you must select the specific register you want to access. The choices will vary depending upon the type of PLC you are addressing.

2. For this example, select ***Timer*** as the file type.
3. Select ***4*** for the file number.
4. Enter ***0*** as the element number. This gives us register 0 of timer file 4.
5. Click ***Next***.



Here you must choose the mnemonic you wish to access.

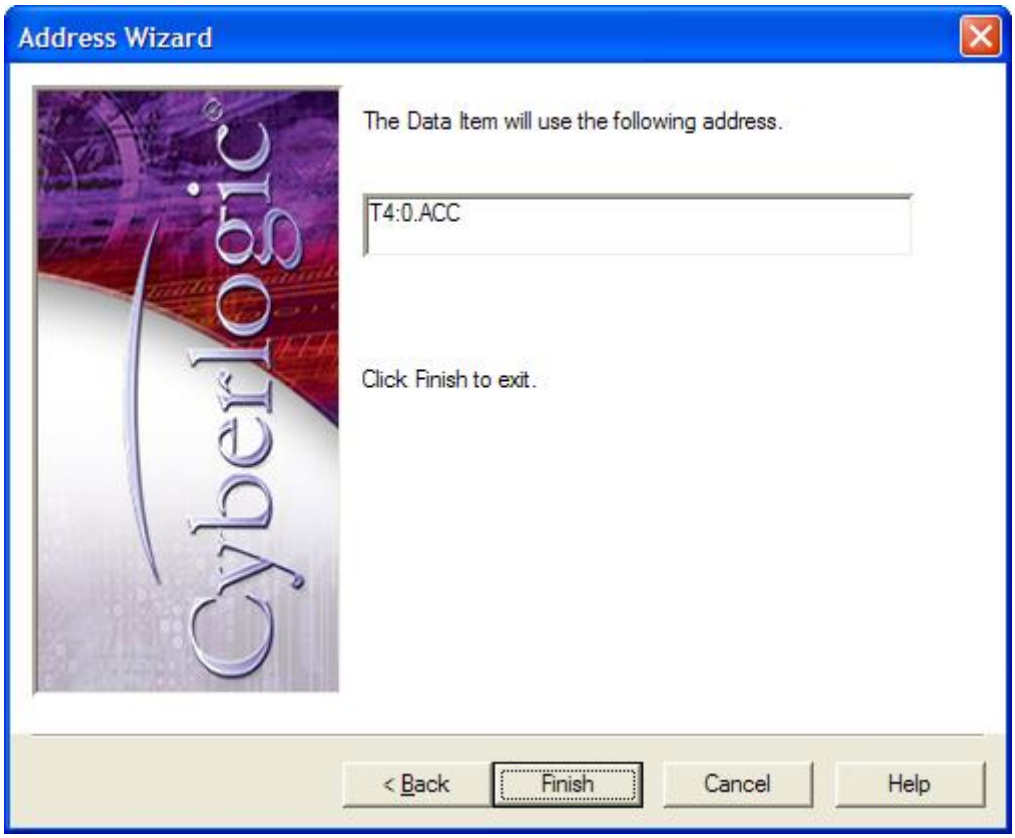
6. Choose ***ACC*** from the drop-down.
7. Click ***Next***.



It is possible to access individual bits or groups of bits within the accumulated value, but we want to access it as a word.

8. Select ***No***.

9. Click ***Next***.



This screen shows the result of your selections. You will access the accumulated value of timer 0 in timer file 4.

10. Click ***Finish***.

Example 3: Arrays of Registers

This example shows how to set up the addressing to allow you to access several registers as an array.

1. From the Welcome screen, click ***Next*** to continue.

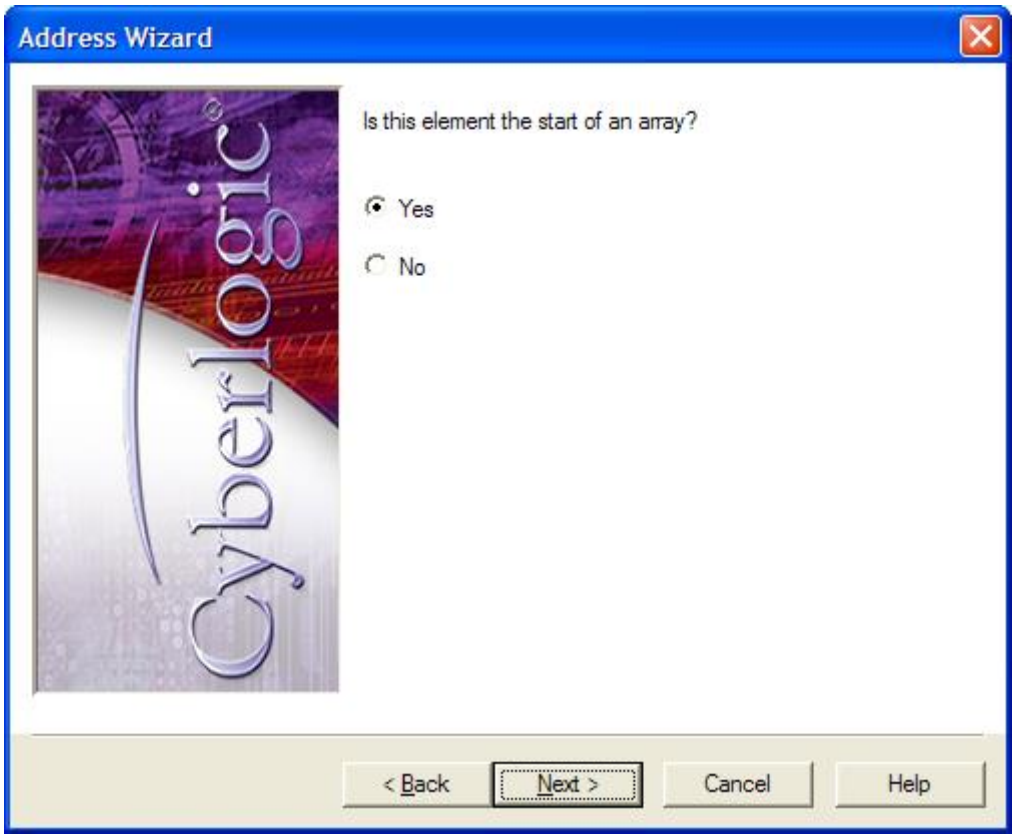
On this screen, you must select the first element of the array, that is, the first register you want to access. The choices will vary depending upon the type of PLC you are addressing.

2. For this example, select ***Integer*** as the file type.

3. Select ***9*** for the file number.

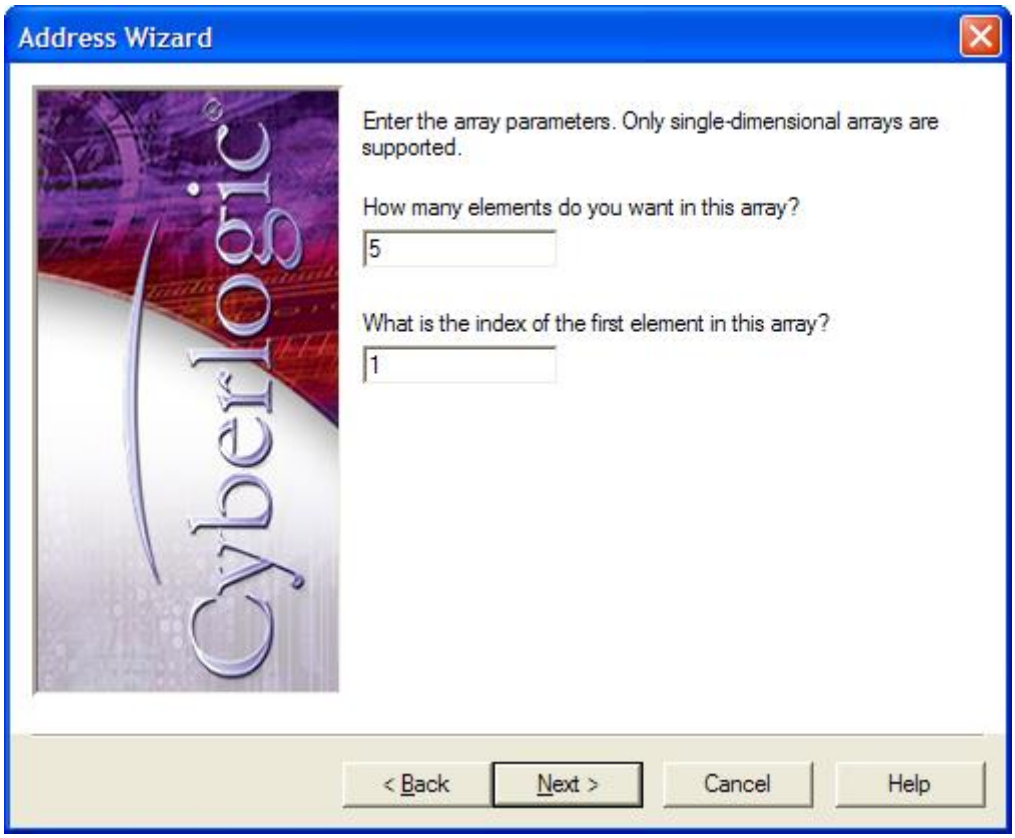
4. Select ***20*** as the element number.

5. Click ***Next*** to continue.



For this example, we want to access the data as an array.

6. Select ***Yes***.
7. Click ***Next***.



Here you must specify the size of the array and how the elements will be referenced.

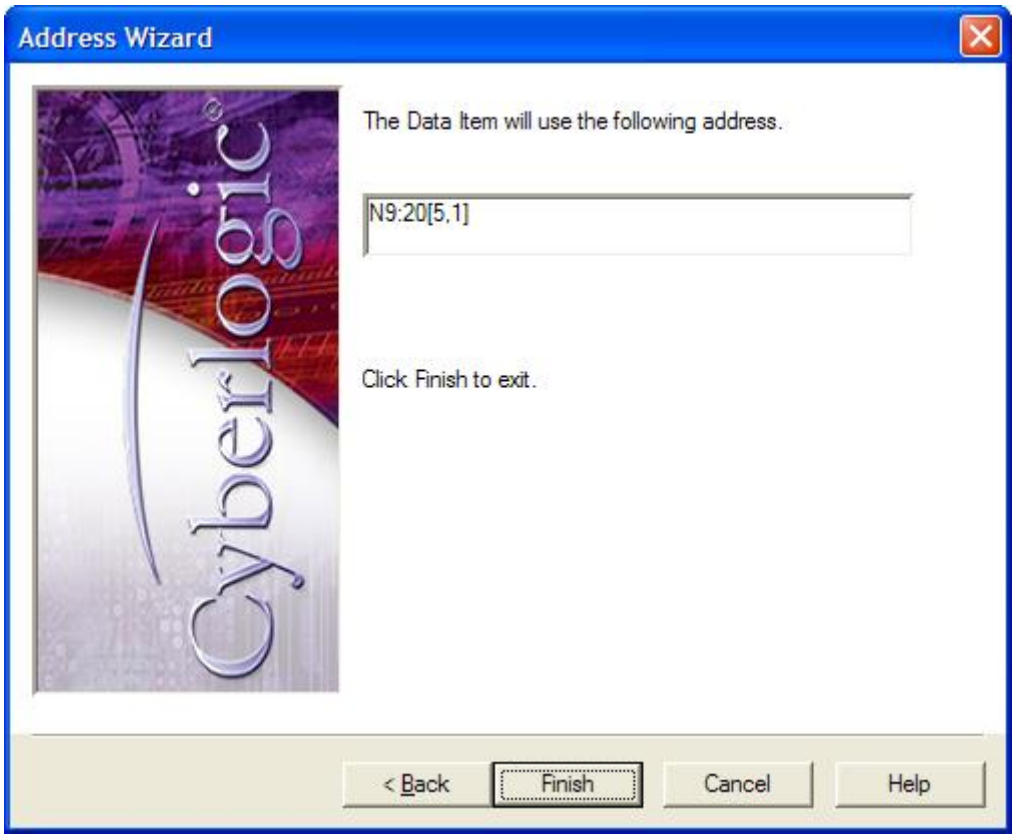
8. Enter ***5*** for the number of elements in the array.

   In this example, the five registers N9:20 through N9:24 will constitute the array.

9. Enter ***1*** for the index of the first element.

   This value specifies how the client will address the array. In this case, the first element's index of 1 will correspond to N9:20. The register N9:21 would then be addressed using index 2, and so on.

10. Click ***Next*** to continue.



This screen shows the result of your selections. The bracketed values show that the array contains five elements and that the index for addressing starts at 1.

11. Click ***Finish***.

APPENDIX D: UNSOLICITED MESSAGE PROGRAMMING

The DHX Driver Agent accepts unsolicited messages from trusted sources that the user identifies by configuring Unsolicited Message Filters. These can be either read or write requests. The read requests are used to read the content of the server status block, while the write requests are used to update data in selected data items. For detailed information on sending communication messages from various PLCs, refer to the appropriate Allen-Bradley documentation.

Read Requests

There are a number of command messages that can be used to read the content of the server status block. Effectively, this agent impersonates PLC-2, PLC-3, PLC-5, SLC-500 and MicroLogix controllers and accepts commands that are normally used for reading data from these processors. The location of the server status block depends upon the specific PLC family. For details, refer to the Server Status Block Tab section.

PLC-2 Family

The DHX driver agent accepts the Unprotected Read commands with the word addresses in the range of 200-205 (octal). Only full word reads are valid.

PLC-3 Family

The DHX driver agent accepts the Word Range Read commands with addresses in the range of N0:0-N0:5.

PLC-5 Family

The DHX driver agent accepts the Typed Read commands with addresses in the range of N7:0-N7:5.

SLC 500 and MicroLogix Family

The DHX driver agent accepts the Typed Logical Read commands with addresses in the range of N7:0-N7:5.

Write Requests

Write requests are intended for updating data items that are configured for unsolicited updates. A single message can update several data items. However, the message must contain data for the entire data item; partial data updates are not allowed. The message commands that you must use to send data depend on the PLC family type of the parent device. The following sections provide some guidelines for all supported message commands.

PLC-2 Family

To update data in the PLC-2 family data items you can use either the Protected Write or the Unprotected Write commands.

PLC-3 Family

To update data in the PLC-3 family data items you must use the Word Range Write command.

PLC-5 Family

To update data in the PLC-5 family data items you must use the Typed Write command.

Caution! When writing to structure sub-elements (e.g. T4:0.ACC) you must send data for the entire structure. Messages containing only sub-element data will not be accepted.

SLC 500 and MicroLogix Family

To update data in the SLC 500/ MicroLogix family data items you must use the Typed Logical Write command.

Caution! When writing to structure sub-elements (e.g. T4:0.ACC) you must send data for the entire structure. Messages containing only sub-element data will not be accepted.

APPENDIX E: USING CONTROLLOGIX PROCESSORS

ControlLogix controllers normally communicate using Cyberlogic's ControlLogix Driver Agent, which is included as part of the DHX OPC Server Suite, DHX OPC Premier Suite and DHX OPC Enterprise Suite. This appendix describes an alternative method that allows you to use the DHX Driver Agent instead.

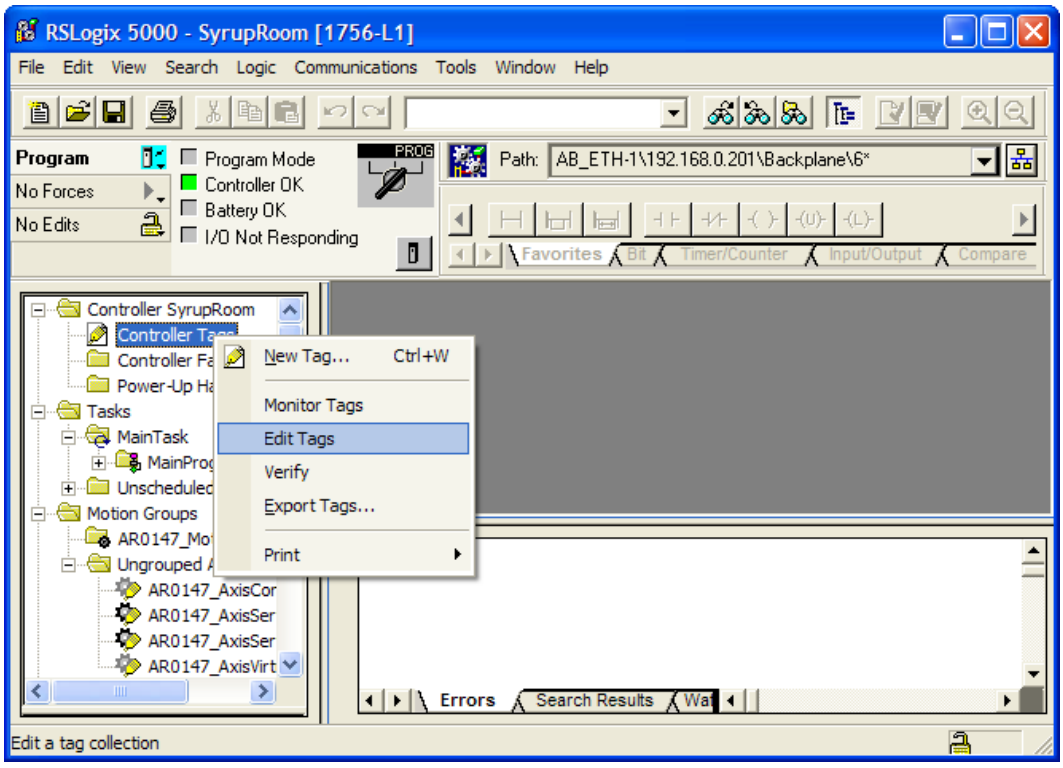
Note Both the DHX and ControlLogix Driver Agents can be used simultaneously to communicate to the same ControlLogix processor.

The DHX Driver Agent does not directly support ControlLogix communications, but it does support PLC-5 communication. You can configure the ControlLogix unit to be compatible with PLC-5 communication, and that allows it to work with the DHX Driver Agent. To do this, you will create a compatibility file in the ControlLogix processor that a PLC-5 can read from and write to. You will then map the compatibility file into a ControlLogix array. This allows you to configure the OPC server to treat the ControlLogix processor like a PLC-5, and communicate with it over Data Highway Plus or Ethernet.

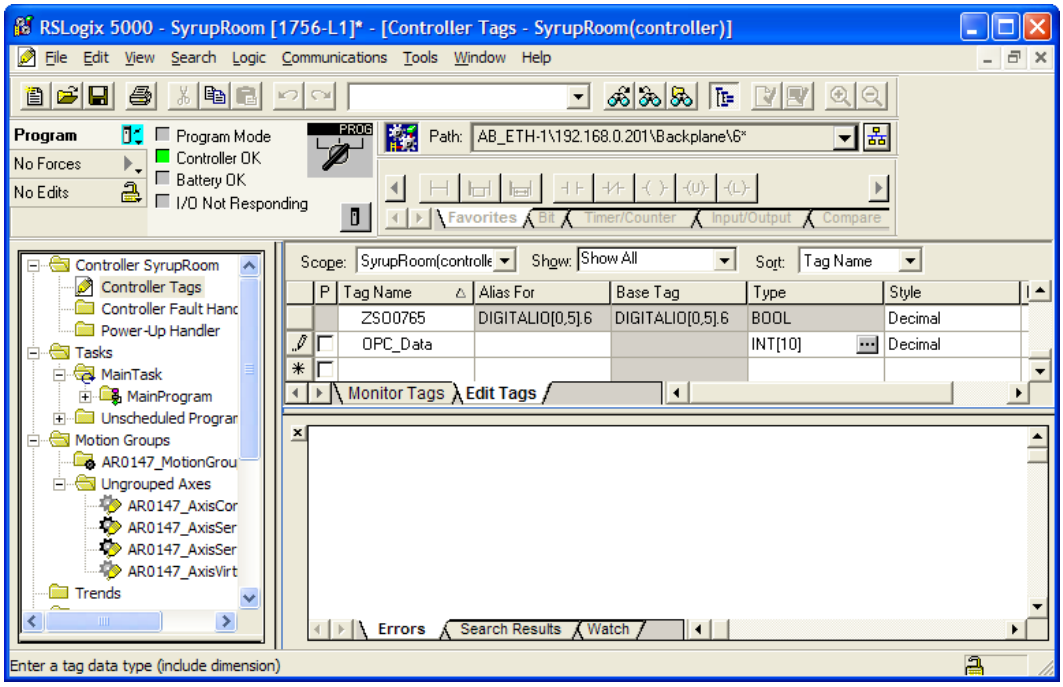
The first step is to to set up the compatibility file in the ControlLogix processor. This is required for both Ethernet and DH+ communication. The remainder of the configuration will depend on the network type.

Configuring the Compatibility File

The compatibility file is required for both network types. It is configured in the ControlLogix program using RSLogix 5000.



1. Right-click on ***Controller Tags*** in the tree in the left pane, and select ***Edit Tags***.

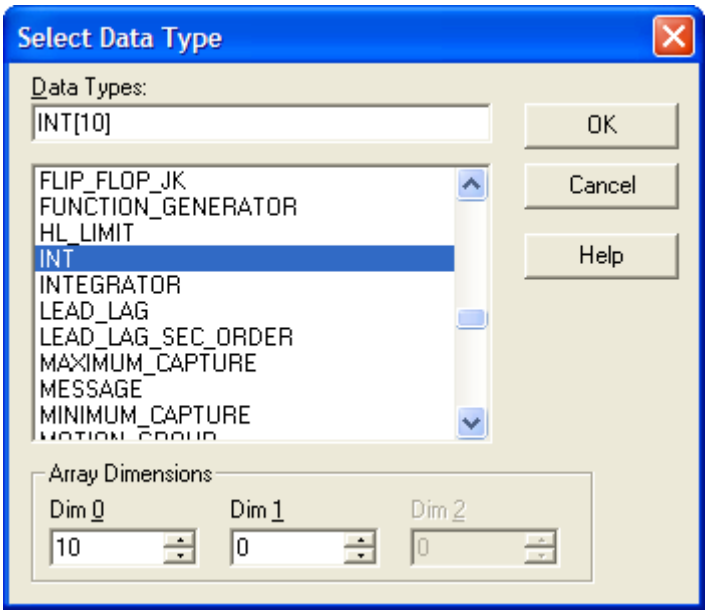


2. Create a tag to hold the data that will be passed to and from the OPC Server.

   The example shown uses an array of ten integers. If you have different data types, you may use more than one tag and create multiple compatibility file mappings.

3. To edit the data type, click its ***Type*** field and then click on the ***ellipsis (...) button***.

   The Select Data Type pop-up will appear.



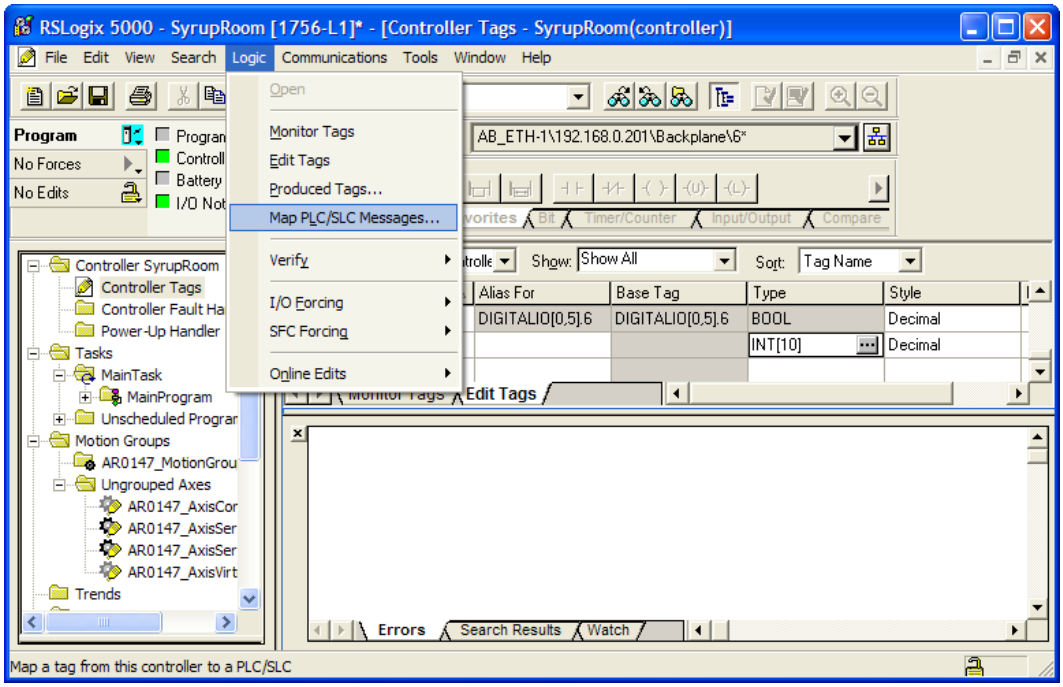
4. Select the value for ***Data Types*** from the list.

5. If applicable, select the dimensions of the array in the ***Array Dimensions*** area.

6. Click ***OK*** to exit and apply the changes.

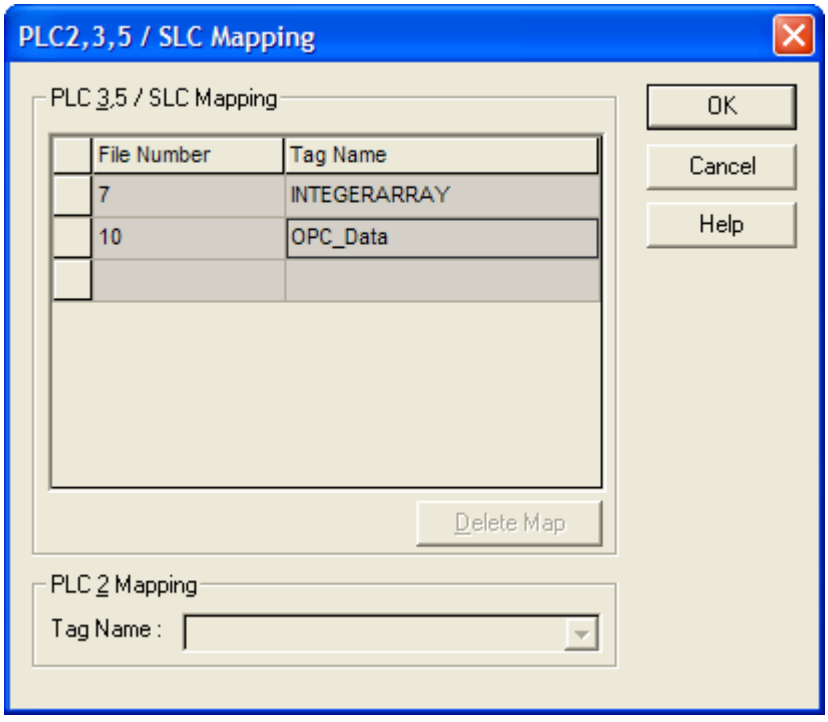
7. When you have finished creating the tags you will need, save the changes.

   You must now configure the PLC/SLC message mapping.



8. Open the ***Logic*** menu and select ***Map PLC/SLC Messages...*** .

   In the mapping window, you must configure the mappings in the ***PLC 3,5 / SLC Mapping*** area.



9. Enter a ***File Number***.

   You may use any valid file number, and need not follow any PLC or SLC conventions for the data types associated with file numbers.

10. In the ***Tag Name*** field, select the desired tag from the drop-down box.

11. Repeat this procedure for any additional mappings you need.

12. Click ***OK***.

13. Save your changes and download them to the ControlLogix processor.

In this example, the OPC Server—or any PLC 3, 5 or SLC—can now read data from and write data to file number 7. Internally, the ControlLogix processor will map these requests to the actual data location, the array OPC_Data.

This completes the ControlLogix portion of the configuration.

The remainder of the process depends on the network type. The choices are:

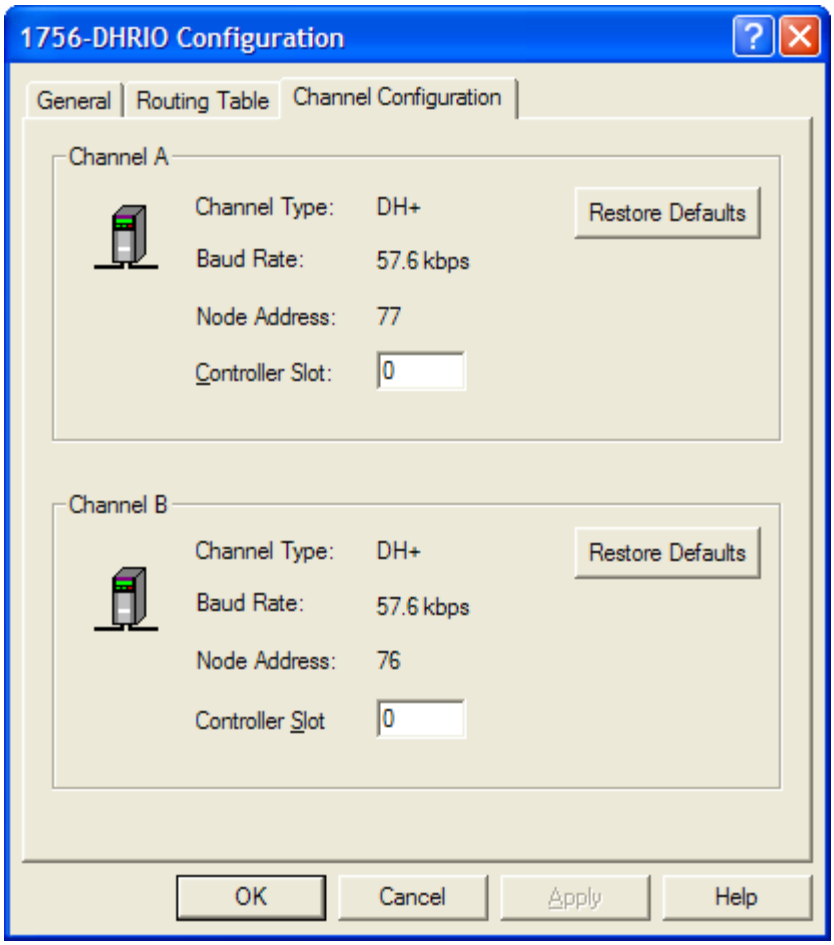
- Data Highway Plus Communication
- Ethernet Communication

Data Highway Plus Communication

Data Highway Plus communication requires a 1756-DHRIO module in the ControlLogix chassis. You must configure this module, and then configure a network node in the OPC server.

Configuring the 1756-DHRIO Module

The ControlLogix chassis uses a 1756-DHRIO module for Data Highway Plus communications. Refer to the 1756-DHRIO documentation for complete instructions on setting up the module. You will configure the module for local DH+ messaging, which means that you can send DH+ messages to only one controller per channel. If you want to communicate with more than one controller in the chassis, you must configure a separate DH+ channel for each.



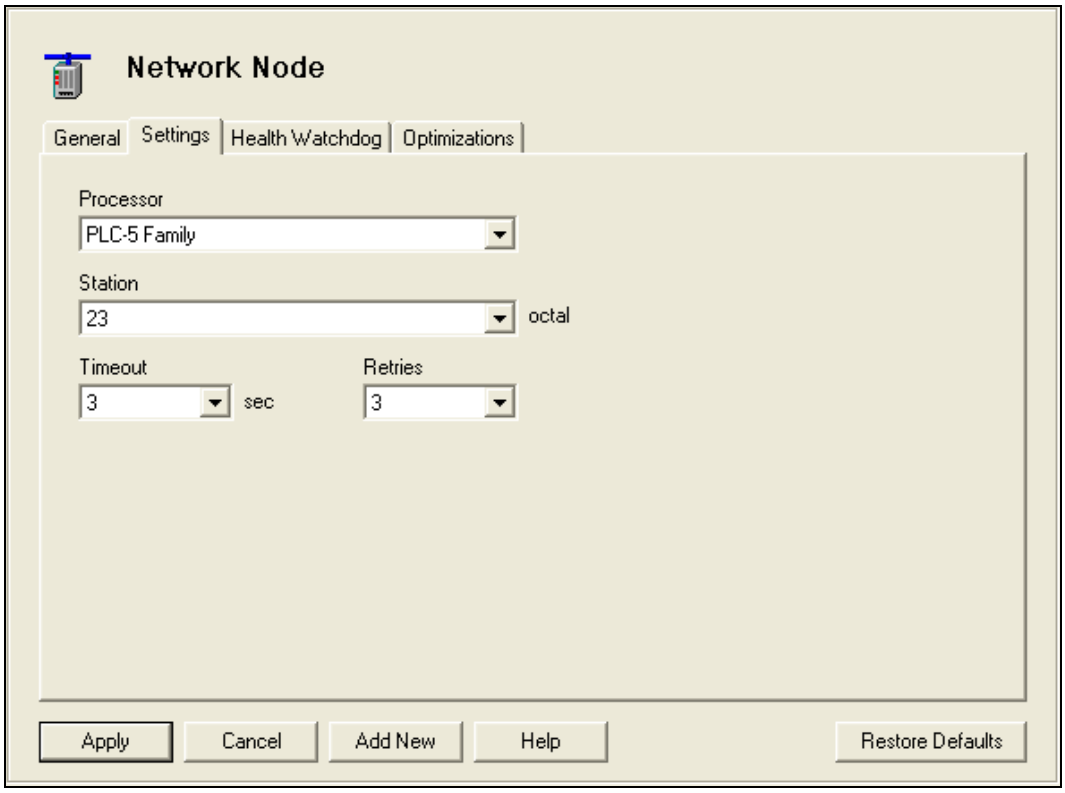
It is not necessary to configure the DHRIO Routing Table.

On the Channel Configuration tab, enter the slot number for the controller you wish to communicate with. The leftmost position in the rack is slot 0. The 1756-DHRIO is a two-channel device, so be sure you are configuring the correct channel.

You must configure a valid controller slot for both channels, even if the second channel will not be used. It is permitted to enter the same controller slot for both channels.

Configuring the OPC Server Network Node

You must now use the OPC Server Configuration Editor to configure a network node for the ControlLogix device. The techniques for creating and editing a network node are explained in the Network Connections section. If you use automatic configuration, the ControlLogix node will be detected and shown as a node of Unknown type.



Configure the node just as you would any other, but in the Processor field of the Network Node, select ***PLC-5 Family***. This will cause the OPC Server to communicate with the ControlLogix processor as though it were a PLC-5, just as required by the compatibility file you configured.

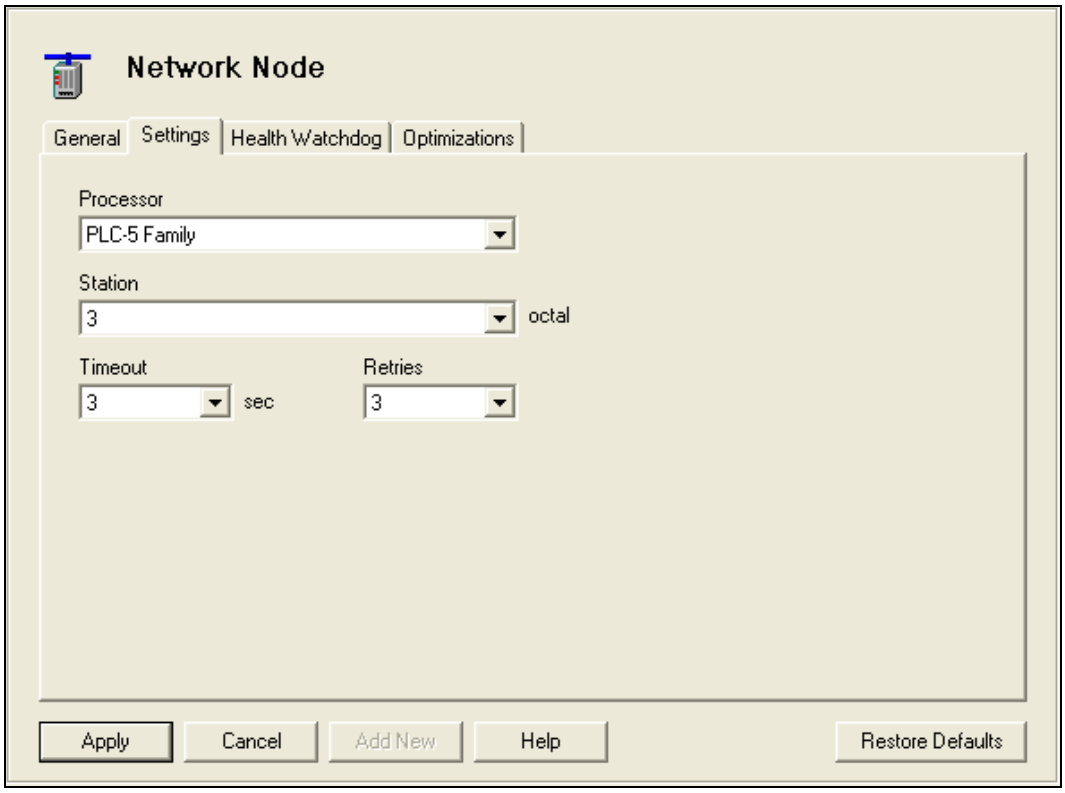
You may now configure devices and data items in the address space just as you would for any other PLC-5. The data items will use PLC-5 style addresses, and will be mapped to the ControlLogix data as you specified in the compatibility file mapping.

In the example used here, you might set up a data item as the array N7:0[10]. When the server reads these from the ControlLogix unit, it will receive the values in the array OPC_Data.

Ethernet Communication

Ethernet communication requires only the configuration of an Ethernet DHX/CIP device in the OPC server. No hardware configuration is needed at the ControlLogix end.

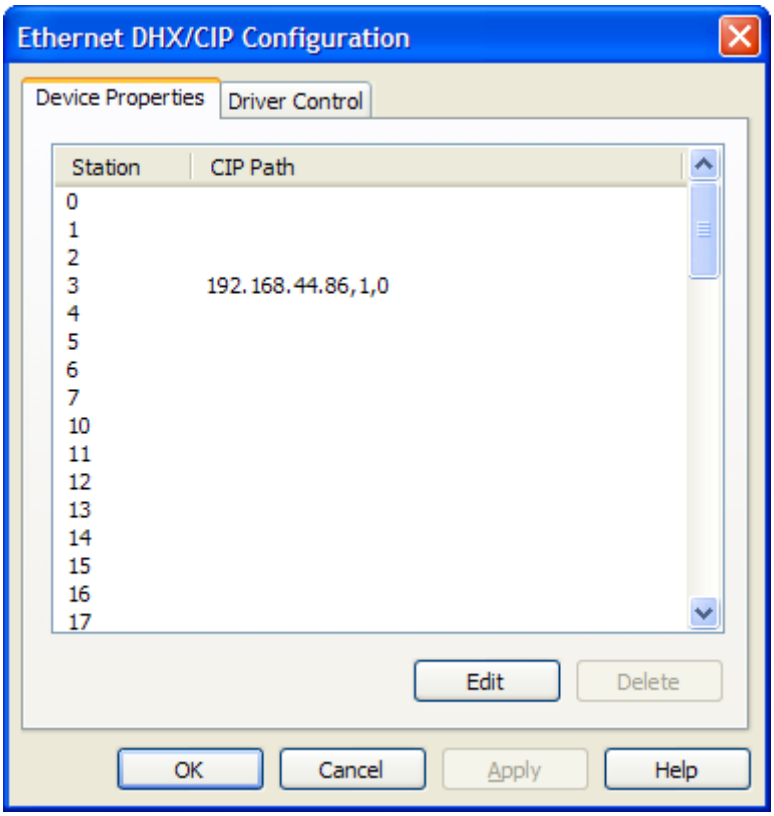
You must now use the OPC Server Configuration Editor to configure a network node for the ControlLogix device. The techniques for creating and editing a network node are explained in the Network Connections section.



Configure the node just as you would any other, but in the Processor field of the Network Node, select ***PLC-5 Family***. This will cause the OPC Server to communicate with the ControlLogix processor as though it were a PLC-5, just as required by the compatibility file you configured.

You may now configure devices and data items in the address space just as you would for any other PLC-5. The data items will use PLC-5 style addresses, and will be mapped to the ControlLogix data as you specified in the compatibility file mapping.

In the example used here, you might set up a data item as the array N7:0[10]. When the server reads these from the ControlLogix unit, it will receive the values in the array OPC_Data.



When you edit the Device Properties of the Ethernet DHX/CIP device, enter the CIP path for the ControlLogix processor in the mapping for the station number you want to use.

APPENDIX F: CONFIGURING MAXIMUM CONCURRENT REQUESTS

The Maximum Network Requests and Maximum Node Requests settings limit the number of transactions that the network connection or network node will process simultaneously. Selecting the proper values for these limits can be crucial in obtaining peak performance from your OPC server. The server uses these limits to determine the best way to balance the network load while ensuring an optimum level of communication with each network node.

Although the default values will work well for most common network layouts, some cases may require a few adjustments. In this appendix, we will discuss how to decide on the proper settings to use, and will provide examples of some typical kinds of configurations.

Two Levels of Limits

Maximum request values are set at both the network connection level and at the network node level.

The maximum network request limit is entered on the Settings Tab of the network connection. This parameter limits the total number of simultaneous transactions for all network nodes on that network connection.

The maximum node request limit is entered on the Optimizations Tab of the network node. This parameter allows you to limit the number of simultaneous transactions for that individual network node. If Unlimited is selected, the number of transactions is limited only by the value set for the network connection.

These parameters interact, so each must be taken into consideration when setting the other. For example, the number of simultaneous transactions actually handled by a network node will never be greater than the lower of the two numbers. Furthermore, the number of simultaneous transactions handled by a network connection will never be greater than the total of the limits set on the nodes on that network.

In addition, you must remember that the resources available to process transactions through the network connection are shared among the network nodes. This further restricts the number of transactions that may be available for communication to a given network node. As the client applications request data from the various network nodes, the server arbitrates these competing requests, allowing them to be processed as resources become available. Thus, the number of transactions being processed by each network node will constantly be changing, but each network node will never process more than its limit, and all of them combined will never process more than the limit for the network connection.

Resources and Performance

The purpose of these parameter settings is to allow the user to allocate system resources and network bandwidth in a way that will yield the best performance. If the concurrent request limit is set too low, throughput will suffer even though there are unused resources available to the server.

However, choosing the maximum setting doesn't always improve performance, and may actually make things worse. There are a number of situations in which you should lower these limits.

- *Bridging to other networks*. If you have two or more different networks bridged together, and the messages must pass through more than one network, you should choose values consistent with the slowest network.
- *Slow PLCs*. Some older PLCs may be able to connect to fast networks, yet they run relatively slowly in their internal processing of messages. Even newer controllers may be bogged down by very large or complex programming. Network nodes like these may not be capable of handling the messages as fast as the network can pass them. A lower limit on simultaneous transactions would keep the PLCs from being overloaded with messages.
- *Other software using the network*. If you have applications other than the OPC server communicating on the network, you may find that the OPC server consumes so much bandwidth that the other applications cannot run satisfactorily. For example, with the OPC server running, your programming software may not be able to connect to the PLC, or uploads and downloads may take excessively long. You may wish to lower the OPC server's limits to prevent the network from saturating and permit the programming software to run.

The goal is to adjust these limits to a level that will allocate the system resources most effectively.

Example 1: Network Node Set to Lower Limit

A network connection is configured for a maximum of 16 network requests and one of its nodes is configured for a maximum of 4.

The limit for that node will be 4 simultaneous transactions, regardless of the fact that the network connection can handle more. This means that there will always be at least 12 concurrent requests available to other network nodes using that network connection.

Example 2: Network Node Set to Unlimited

A network connection is configured for a maximum of 8 network requests and one of its nodes is configured as Unlimited.

The maximum number for that node would be 8, that is, everything the network connection can handle. If the network connection limit is increased to 16, then the limit for that network node would increase as well.

Example 3: Interaction Between Multiple Network Nodes

The network connection is configured for a maximum of 20 network requests. It serves five network nodes, each with a limit of 8.

The total of the limits for the five network nodes is then 40, twice what the network connection can handle. This simply means that it will not be possible run the limit of 8 requests on all five network nodes at the same time.

In this configuration, no more than two network nodes could run at their limit of 8 simultaneous transactions, and occasionally one or two might reach that limit. However, the server will always try to spread the load equally among all nodes, so most of the time all nodes would be running at about 4 simultaneous transactions.

Example 4: Network Nodes with Varying Speeds

The network connection is configured for a maximum of 10 network requests. It serves three network nodes, one of which is limited to a maximum of 8 node requests, and the other two limited to 2.

The benefit of this configuration is that it avoids the situation where the slower nodes are overwhelmed with transactions they cannot handle, while the faster node is deprived of transactions that it could handle. When the slower nodes are each handling two transactions, the remaining 6 that the network can handle will be available to the faster node.

Example 5: Bridging to a Slower Network for All PLCs

The network connection provides connectivity to an Ethernet network, but all of the messages pass through a bridge to a Data Highway Plus network before reaching the PLCs.

The recommended range for the maximum network requests on an Ethernet network is 16-32, but the range for DH+ is 8-16. In this situation, the slower network determines the throughput, so the correct range to use is the lower of the two. You should select a value between 8 and 16.

Example 6: Bridging to a Slower Network for Some PLCs

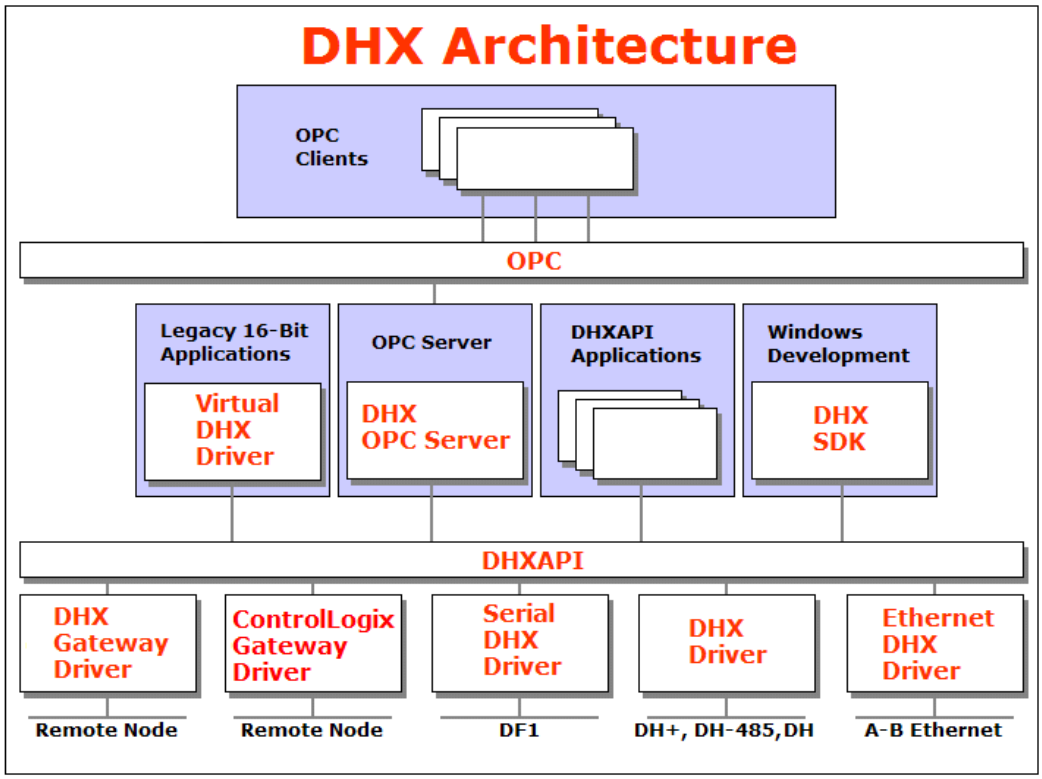
The network connection provides connectivity to an Ethernet network, and some of the PLCs are directly connected to Ethernet. For others, the messages must pass through a bridge to reach the PLCs via a serial DF1 connection.

The recommended range for the maximum network requests on an Ethernet network is 16-32, but the range for DF1 is 2-4. You can set up the network connection for the Ethernet range of 16-32, and do the same for the network nodes that are directly connected to the Ethernet, perhaps specifying them as Unlimited. For the network nodes that use DF1, you must set the maximum node requests in the 2-4 range, to accommodate the slower connection they use.

APPENDIX G: DHX ARCHITECTURE AND COMPANION PRODUCTS

The DHX Driver is part of the Cyberlogic DHX family. This family consists of several well-integrated products that provide connectivity for Data Highway, Data Highway Plus, DH-485, ControlNet and Ethernet networks in distributed environments.

This section illustrates the layout of the DHX architecture. It includes a description of each DHX component along with suggested methods for employing them to support Allen-Bradley networks.



The DHX architecture presents a consistent framework to address different connectivity needs.

DHX Driver

The DHX Driver provides connectivity between Windows-based applications and interface adapter cards from Allen-Bradley and SST. A few of the many cards supported are the 1784-PKTX and 1784-PCMK from Allen-Bradley, as well as the SST DHP-PCI and 5136-SD-PCI from SST. These provide communication over Data Highway, Data Highway Plus and DH 485.

The kernel mode device driver of the DHX Driver has exceptional performance and stability. It operates in either interrupt or polled mode and fully implements all network features, including solicited and unsolicited communication. The high performance native API (DHXAPI) of the DHX Driver takes full advantage of the event-driven, multitasking, multithreaded features of Windows operating systems.

The driver includes the DHX Gateway Server for remote access by the DHX Gateway Driver and is fully compatible with all other components of the DHX family.

The DHX Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

Ethernet DHX Driver

The Cyberlogic Ethernet DHX Driver emulates Data Highway Plus over the Ethernet TCP/IP protocol. It supports most DHXAPI and 6001-F1E-compatible software, providing instant access to Ethernet TCP/IP compatible devices without code modifications. It is compatible with all Ethernet cards supported by Windows.

The driver includes the DHX Gateway Server for remote access by the DHX Gateway Driver and is fully compatible with all other components of the DHX family.

The Ethernet DHX Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

Serial DHX Driver

The Cyberlogic Serial DHX Driver provides connectivity to full-duplex DF1-compatible devices through standard serial COM ports. These devices include the 1770-KF2, 1785-KE, 1770-KF3 and 1770-KFC15 interface modules for Data Highway, Data Highway Plus, DH-485 and ControlNet, as well as direct connection to devices with full-duplex DF1-compatible ports. The Serial DHX Driver supports both the DF1 BCC and DF1 CRC-16 protocols.

The driver includes the DHX Gateway Server for remote access by the DHX Gateway Driver and is fully compatible with all other components of the DHX family.

The Serial DHX Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

DHX Gateway Driver

The DHX Gateway Driver lets applications use DHX devices on remote DHX Gateway Server nodes as though they were on the local system. The client system running the DHX Gateway Driver must be a Windows node connected over a standard LAN to another system running the DHX Gateway Server. It can then access the Data Highway, Data Highway Plus, DH-485 and ControlNet networks that are connected to the server node.

For example, the DHX Gateway Driver provides complete DHX Driver functionality to the client node applications. An interface adapter, such as a 1784-PCMK card, is not required on the client node. DHX Gateway Driver nodes can communicate with multiple remote servers and all Windows-compatible TCP/IP networks are supported.

The DHX Gateway Driver is compatible with all other components of the DHX family.

The DHX Gateway Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

ControlLogix Gateway Driver

The ControlLogix Gateway Driver lets applications access Data Highway Plus networks from a remote location through a ControlLogix gateway module. With this driver, a remote system can communicate over a standard Ethernet network to a ControlLogix chassis containing a 1756-DHRIO module. That module then acts as a gateway to a Data Highway Plus network. This allows the remote system to access the PLC-5s, SLC-500s and any other devices on the Data Highway Plus network as though it were connected directly to that network.

The ControlLogix Gateway Driver is fully compatible with all other components of the DHX family.

The ControlLogix Gateway Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

Virtual DHX Driver

The Virtual DHX Driver allows 16-bit DOS and Windows applications using 1784-KT/KTX interface adapters to run concurrently with 32-bit applications on the same computer. It allows multiple 16-bit applications and multiple instances of a single 16-bit application to run under the latest Windows operating systems. By emulating the physical 1784-KT/KTX

adapters, the Virtual DHX Driver will work with all legacy software, regardless of which DOS driver is used.

If your computer uses Windows 7 or the 64-bit edition of any Windows version, refer to Cyberlogic Knowledge Base article *KB2010-02 Running 16-Bit Applications* for important information on using the Virtual DHX Driver on your system.

The Virtual DHX Driver is fully compatible with all DHX components and requires at least one of these drivers to operate:

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

The Virtual DHX Driver is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite
- DHX Driver Suite

DHX OPC Server

The Cyberlogic DHX OPC Server connects OPC-compliant clients to Data Highway, Data Highway Plus, DH-485, ControlNet and Ethernet networks. It supports the latest OPC Data Access and OPC Alarms and Events specifications and uses the DHX drivers for connectivity to Allen-Bradley networks.

The DHX OPC Server supports multiple, priority-based access paths for reliable, redundant communications. It also supports both solicited and unsolicited communications and uses an advanced transaction optimizer to guarantee minimum load on your networks. With only a couple of mouse clicks, the DHX OPC Server will automatically detect and configure the attached networks and node devices. Other noteworthy features include DirectAccess, Data Write Protection and Health Watchdog.

The DHX OPC Server is included in the following products:

- DHX OPC Enterprise Suite
- DHX OPC Premier Suite
- DHX OPC Server Suite

DHX SDK

Software developers can use the DHX Software Development Kit to provide connectivity to Data Highway, Data Highway Plus, DH-485, Ethernet and ControlNet networks from their 32-bit and 64-bit C/C++/C# applications.

The SDK supports 6001-F1E and Cyberlogic's high-performance DHXAPI and DHXAPI.Net interfaces. The 6001-F1E interface is an excellent bridge for developers who would like to port their 16-bit applications to the latest Windows environments. Developers of new applications can use any of the three interfaces. For a complete 6001-F1E specification, contact any Allen-Bradley distributor.

Since all DHX family drivers are built on the same DHX architecture, applications developed with the DHX SDK can be used with all DHX family drivers and can execute under all current Windows operating systems.