OPTOSERVER USER'S GUIDE

Form 722-130822—August, 2013

OPTO 22
Automation made simple.

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

OptoServer User's Guide
Form 722-130822—August, 2013

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or later are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Wired+Wireless controllers and brains and N-TRON wireless access points are licensed under one or more of the following patents: U.S. Patent No(s). 5282222, RE37802, 6963617; Canadian Patent No. 2064975; European Patent No. 1142245; French Patent No. 1142245; British Patent No. 1142245; Japanese Patent No. 2002535925A; German Patent No. 60011224.

Opto 22
Automation Made Simple

Table of Contents

Welcome to OptoServer

OptoServer™ is Opto 22's OPC 1.0-compliant, DDE, and NetDDE server for Microsoft® Windows® systems. Part of the Opto 22 FactoryFloor® suite of software. OptoServer is the client-server application that supports the bidirectional flow of data between PC clients on the FactoryFloor network and controllers on the Opto 22 controller network.

OptoServer works seamlessly with OptoDisplay™ and OptoControl™ as stand-alone applications or within a complex client-server architecture. OptoServer can also be used to integrate Opto 22 control hardware and software solutions with Microsoft products, third-party packages, and custom applications you can create using development tools such as Microsoft Visual Basic® or Visual C++®.

About This Guide

This guide assumes that you are already familiar with Microsoft Windows on your personal computer, including how to use a mouse, standard commands, and menu items to open, save, and close files. If you are not familiar with Windows or your PC, refer to the documentation from Microsoft and your computer manufacturer.

This guide also assumes that you are familiar with Dynamic Data Exchange (DDE) and OLE for Process Control (OPC) if you choose to use OptoServer with DDE-aware and OPC-compliant clients. For information on DDE, consult www.microsoft.com/dev. To learn about OPC, contact the OPC Foundation at www.opcfoundation.org.

Here's what is in this user's guide:

Chapter 1: Quick Start—Two quick demonstrations: how to set up OptoServer to run with an OptoDisplay client and to run with a DDE-aware client.

Chapter 2: What Is OptoServer?—Overview of what OptoServer does and its value to your system, as well as client/server basics.

Chapter 3: OptoServer and DDE-Aware Clients—Details of message transactions between DDE-aware applications and how to use OptoServer with sample programs for Microsoft Visual Basic.

Chapter 4: OptoServer and OPC—Overview of OPC and how to use OptoServer with a sample OPC client.

Appendix A: OptoServer Troubleshooting—Tips for resolving problems you may encounter while running OptoServer.

Appendix B: OptoServer Files—A list of OptoServer files in the OptoServer directory.

Appendix C: FactoryFloor Glossary—Terms used in the FactoryFloor suite of products.

OptoServer Index—Alphabetical list of key words and the pages they are located on.

Document Conventions

The following conventions are used in this document:

- Italic typeface indicates emphasis and is used for book titles. (Example: "See the *OptoDisplay User's Guide* for details.")
- Names of menus, commands, dialog boxes, fields, and buttons are capitalized as they appear in the product. (Example: "From the File menu, select Print.")
- File names appear either in all capital letters or in mixed case, depending on the file name itself. (Example: "Open the file TEST1.txt.")
- Key names appear in small capital letters. (Example: "Press SHIFT.")
- Key press combinations are indicated by plus signs between two or more key names. For example, SHIFT+F1 is the result of holding down the shift key, then pressing and releasing the F1 key. Similarly, CTRL+ALT+DELETE is the result of pressing and holding the CTRL and ALT keys, then pressing and releasing the DELETE key.
- "Click" means press and release the left mouse button on the referenced item. "Right-click" means press and release the right mouse button on the item.
- Menu commands are referred to with the Menu?Command convention. For example, "File?Open Project" means to select the Open Project command from the File menu.
- Numbered lists indicate procedures to be followed sequentially. Bulleted lists (such as this one) provide general information.

Other FactoryFloor Resources

Documents and Online Help

To help you understand and use the FactoryFloor suite of products, the following resources are provided:

- **Online Help** is available in OptoControl, OptoDisplay, OptoServer, and most of the OptoUtilities. To open online Help, choose Help?Contents and Index in any screen.
- *OptoControl User's Guide, OptoDisplay User's Guide,* and *OptoServer User's Guide* give step-by-step instructions for using each of these products. The *OptoServer User's Guide* binder also contains a master **FactoryFloor Glossary,** which defines terms for all FactoryFloor products.

  Online versions (Adobe® Acrobat® format) of these and other FactoryFloor documents are available from the Help menu in your FactoryFloor application. To view a document, select Help?Manuals, and then choose a document from the submenu.
- *OptoControl Command Reference* contains detailed information about each command (instruction) available in OptoControl.

- Two **quick reference cards**, *OptoControl Commands* and *Beginner's Guide to OptoControl Commands*, are located in the front pocket of the *OptoControl Command Reference*.
- FactoryFloor resources are also available on the Opto 22 Web site at factoryfloor.opto22.com. You can conveniently access this and other sections of the Opto 22 Web site using the Help menu in your FactoryFloor application. Select Help?Opto 22 on the Web, and then select an online resource from the submenu.

Product Support

If you have any questions about FactoryFloor, you can call, fax, or e-mail Opto 22 Product Support.

Phone:	800-TEK-OPTO (800-835-6786) 951-695-3080 (Hours are Monday through Friday, 7 a.m. to 5 p.m. Pacific Time)
Fax:	951-695-3017
Email:	support@opto22.com
Opto 22 website:	www.opto22.com

NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.

When calling for technical support, be prepared to provide the following information about your system to the Product Support engineer:

- Software and version being used
- Controller firmware version
- PC configuration (type of processor, speed, memory, operating system)
- A complete description of your hardware and operating systems, including:
  - jumper configuration
  - accessories installed (such as expansion daughter cards)
  - type of power supply
  - types of I/O units installed
  - third-party devices installed (e.g., barcode readers)
- Specific error messages seen.

Installing OptoServer

OptoServer installation is easy and quick. Insert the FactoryFloor CD in your CD-ROM drive, and the installation wizard should appear. If the wizard does not appear, start Windows Explorer and navigate to your CD-ROM drive. Double-click Setup.exe to begin installation.

If you have trouble installing OptoServer or need 3.5-inch disks rather than a CD, contact Opto 22 Product Support at 800/835-6786 or 951/695-3080.

System Requirements

Installation Requirements

Here's what you need to install and run FactoryFloor_app_name:

- A computer with at least the minimum processor required for your version of Microsoft® Windows® (1 GHz Pentium®-class or better recommended). Additional computer requirements include:
  - Ethernet capability, if using an M4-series controller with M4SENET-100 Ethernet adapter card.
  - An RS-232 serial port and serial cable, for downloading firmware updates to a controller.
- Microsoft Windows XP or Windows 2000® workstation operating system with the most recent service packs.
- At least 128 MB RAM (256 MB recommended)
- At least 125 MB of available hard drive space
- VGA or higher resolution monitor (Super VGA recommended)
- Mouse or other pointing device
- Installed Windows printer (optional).

For OptoDisplay and OptoControl Users

If you're planning to use an OptoDisplay or OptoControl client with OptoServer, the minimum versions are OptoDisplay 3.1 and OptoControl 3.1.

Network Requirements

TCP/IP network protocol is required if OptoServer is used with OptoControl, OptoDisplay, or other applications using the OptoCom.dll (such as Microsoft Visual Basic applications you can write yourself). The networking connection uses RPC over TCP/IP.

IMPORTANT: If you have an ARCNET card in your PC at the time you install a Windows 95, Windows 98, or Windows NT operating system, make sure a network driver doesn't get installed on the card. Windows 95, Windows 98, and Windows NT operating systems automatically try to install a network driver on an ARCNET card. Allowing the driver to be installed can produce unpredictable results from OptoServer. For more information, see the Application Note on our Web site, www.opto22.com.

Additional Software Requirements

Verify that the minimum version of OptoCom.dll is 3.1. This driver is used by any program that needs to communicate to a controller.

Additional Hardware Requirements

OptoServer requires some additional hardware as well as some firmware. Typically, you need:

- Opto 22 controller
- Opto 22 I/O units (Opto 22 brains plus SNAP, G4, or G1 I/O modules)

To communicate with a controller, the following additional hardware is required:

- One of the following communication methods:
  - Standard RS-232 serial port
  - Ethernet card
  - Contemporary Controls PCA66 Series ARCNET card
  - Contemporary controls PCI20 ARCNET cards (PCI20-CXS, PCI20-FOG-ST, or PCI20-485)
  - Opto 22's AC24AT, AC37, AC47, or AC422AT adapter cards
- Serial cables, or ARCNET coaxial cables and hubs, for multidrop connections from the PC to the controllers. Opto 22 recommends the Contemporary Controls Mod Hub series of active hubs. The appropriate expansion cards are:
  - EXP-CXS Coax Star
  - EXP-FOG-ST Fiber ARCNET
  - EXP-485 Twisted-pair ARCNET (DC coupled)

NOTE: If you are using OptoRuntimePC for your controller, you do not need any additional hardware to communicate with it, although you may require an adapter card to communicate with your I/O units. Also, if you are using a G4LC32ISA or G4LC32ISA-LT controller, you can communicate with it through the ISA bus in your PC.

OPC

No additional hardware is necessary for OPC.

DDE and NetDDE

OptoServer is compatible with DDE and NetDDE. NetDDE is required for clients connecting to OptoServer across a network. OptoServer also supports Fast DDE and Advanced DDE.

Installing on Windows NT

Windows NT users must have write access to the registry and the Windows\System32 directory to successfully install and use this product.

If FactoryFloor has never been installed on the computer before, you must reboot your Windows NT system after the first time you make changes to communication settings for any port. If an older version of FactoryFloor was previously installed on the computer, however, one or more controllers will already be defined, and it is not necessary to reboot the system.

Firmware Requirements

Firmware is loaded on your controller so that you can download and run OptoControl strategies. If your controller's firmware is not at the required release number, you'll receive an error message. You can use the OptoTerm utility (explained in the Troubleshooting appendix of the *OptoControl User's Guide*) to download the firmware to the controller.

NOTE: If you have a non-flash controller, you need to contact Opto 22 Product Support for an EEPROM upgrade.

Quick Start

Introduction

In this chapter, we'll explain how to quickly get OptoServer up and running on a PC in two ways: first, communicating with an OptoDisplay client in a networked environment, and second, communicating with a DDE-aware application.

The two Quick Start procedures differ because OptoServer has two executable files for communicating with client applications:

- Communication Data Server (OptoCds.exe) is used for OptoDisplay, OptoControl, and other programs that use the OptoCom.dll for communication.
- OptoServer (Mds.exe) is used for DDE-aware and OPC-compliant applications.

We'll discuss these differences in more detail in Chapter 2.

In This Chapter

Communicating with an OptoDisplay Client

In this first Quick Start section we will set up OptoServer to provide information to an OptoDisplay client. In a typical setup, OptoServer and OptoDisplay are running on different computers. For this example, however, we'll run OptoServer and OptoDisplay on the same computer so we can easily demonstrate the configuration process. This section guides you through the steps to monitor the Cookie Controller from OptoDisplay as if OptoDisplay were talking to OptoServer on another computer. Briefly, these steps are:

- Use OptoControl to download the strategy to the controller and verify communication. The OptoControl strategy used is the cfactory.cdb example that comes with OptoServer.
- Start the Communication Data Server (OptoServer) so that it is ready to be accessed by OptoDisplay.
- Configure OptoDisplay to communicate to a server-based controller rather than a direct, or local, controller. The OptoDisplay project used is cfactory.mmi.

- Start OptoDisplay Runtime.

Downloading the Strategy to the Controller and Verifying Communication

1. Start OptoControl by clicking the Start button and selecting Programs > Opto 22 > FactoryFloor 4.0 > OptoControl > OptoControl.
2. From the File menu, choose Open Strategy. Navigate to Opto 22\Shared\Cookies\Control and open the file cfactory.cdb.
3. Configure the strategy for your controller's connection, address, and port settings, and enter "Cookie Controller" as the controller's name.

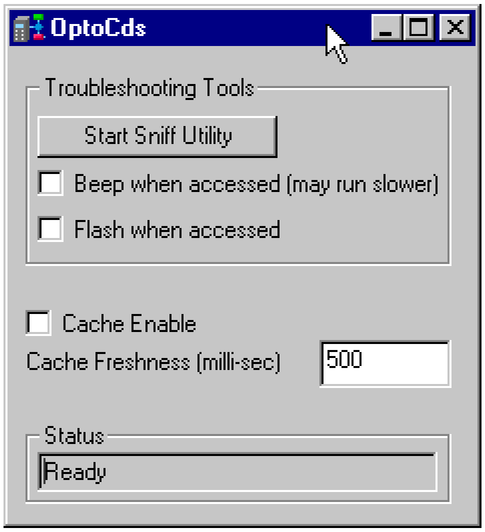
   If you need help, see "Configuring Controllers" in the *OptoControl User's Guide*.
4. Compile the program by selecting Compile > Compile All.
5. Change to Debug mode and run the program.
6. Verify that the controller is communicating with your PC. Observe whether the controller's host port LEDs (ARCNET, COM0, etc.) flash in response to messages from the PC.

   If your controller is responding to your PC with no errors, you're ready to move on to the next step. If you think you're having problems getting your controller to respond to the OptoControl strategy, make sure the configuration setup and cable wiring between the controller and PC are correct. Consult the *OptoControl User's Guide* and your controller manual to help you troubleshoot the problem.
7. When the controller is communicating with the PC, leave the strategy running and exit OptoControl.

Starting Communication Data Server (OptoServer)

1. Click the Start button and select Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > Communication Data Server.

   The Communication Data Server window appears:



□ Beep when accessed (may run slower)

□ Flash when accessed

□ Cache Enable

Cache Freshness (milli-sec) 500

Ready

2. Click the minimize button 🗕 to reduce the Communication Data Server window to an icon.

Configuring OptoDisplay

Next, we'll configure the OptoDisplay sample program so it uses OptoServer for its data updates. To do so, we must change OptoDisplay's PC communication port (for example, COM1, COM2, ARCNET) from its direct controller hardware connection to the networked PC that is running OptoServer.

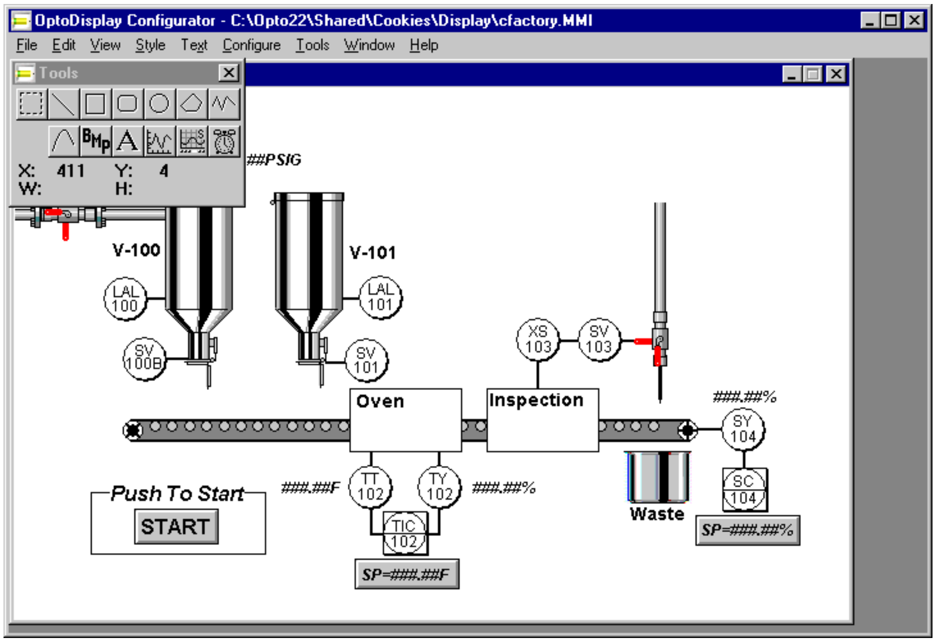
1. Start OptoDisplay Configurator by clicking the Start button and selecting Programs > Opto 22 > FactoryFloor 4.0 > OptoDisplay > OptoDisplay Configurator.

   The OptoDisplay Configurator main window appears.

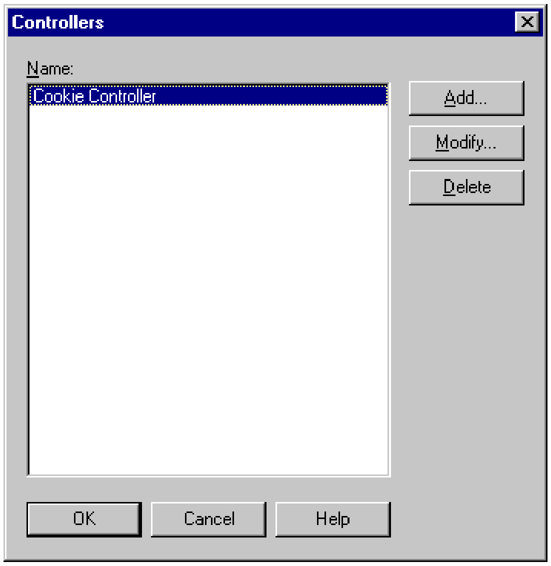
2. Select File > Open Project.

3. In the Open Project dialog box, navigate to Opto 22\Shared\Cookies\Display and open cfactory.mmi.

   The OptoDisplay project is loaded and its graphics appear in the main draw window:



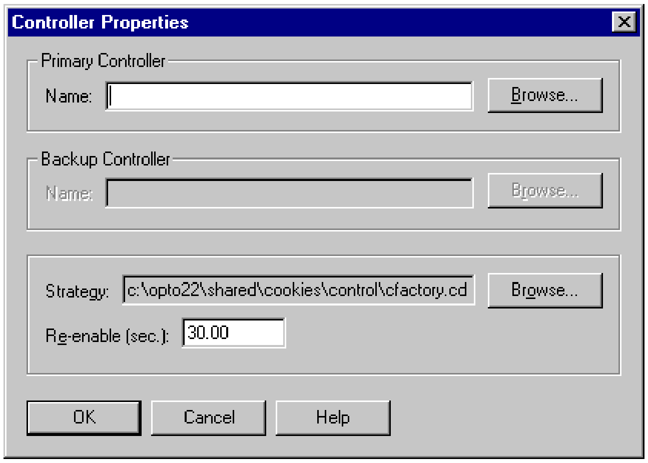
4. Choose Configure > Controller(s).



Name:
Cookie Controller

5. Choose one of the following:
   - If Cookie Controller appears in the list, click Modify so you can change its configuration. Skip to step 7.
   - If Cookie Controller does not appear in the list, click Add.
6. In the Strategy File Name Selection dialog box, select Shared\Cookies\Control\cfactory.cdb.

   The Controller Properties dialog box appears. The controller name may or may not be shown:

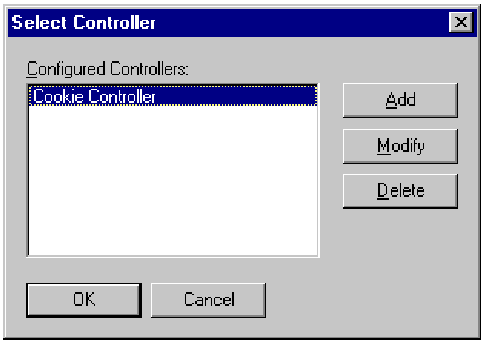


Strategy: c:\opto22\shared\cookies\control\cfactory.cd

Re-enable (sec.): 30.00

7. In the Controller Properties dialog box, click the Browse button next to Primary Controller.

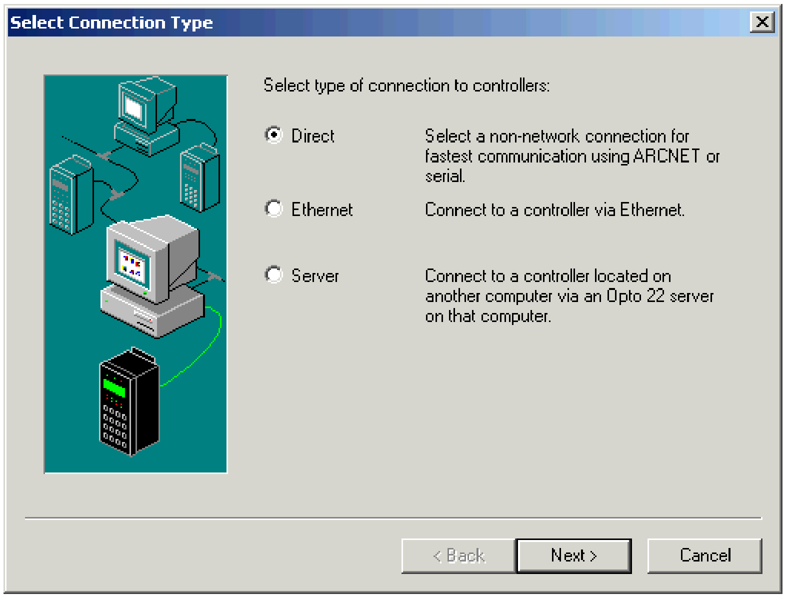
   The Select Controller dialog box appears:



Configured Controllers:
Cookie Controller

8. Select Cookie Controller from the list and click Modify.

   The Select Connection Type dialog box appears, with Direct shown as the type of connection. We're going to change the connection.



Select Connection Type

Select type of connection to controllers:

● Direct — Select a non-network connection for fastest communication using ARCNET or serial.

○ Ethernet — Connect to a controller via Ethernet.

○ Server — Connect to a controller located on another computer via an Opto 22 server on that computer.

9. Click Server and then click Next.

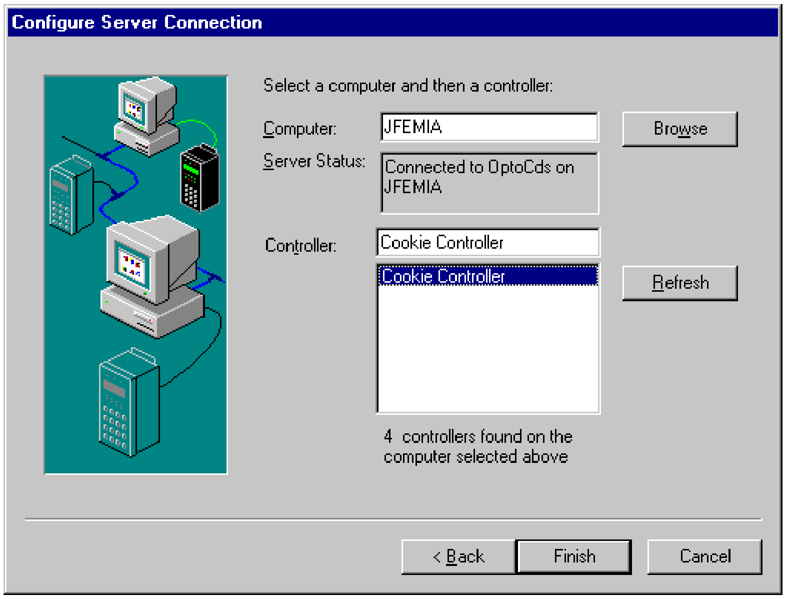
This option connects the OptoDisplay project to a controller located on another computer on the same network.

10. In the Configure Server Connection dialog box, click the Browse button and locate the computer where the OptoServer program (Communication Data Server) is running.

For this example it's the same computer; but usually it is a different computer on the network. Communication Data Server must be running on the computer you choose. When you choose a computer, a list of controllers attached to the computer appears in the list.

11. Highlight the controller you want to use to update the OptoDisplay strategy (in this case, Cookie Controller).

Its name appears in the Controller field, and the dialog box looks something like this:



Configure Server Connection

Select a computer and then a controller:

Computer: JFEMIA

Server Status: Connected to OptoCds on JFEMIA

Controller: Cookie Controller

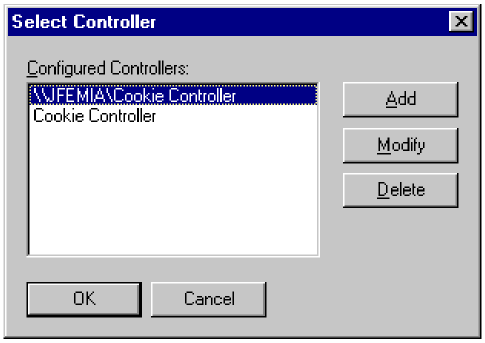
Cookie Controller

4 controllers found on the computer selected above

You may need to click Refresh to see the controller names.

12. Click Finish.

    Notice that a new controller name appears in the Configured Controllers list. It is the name of the server computer with the controller's name appended:



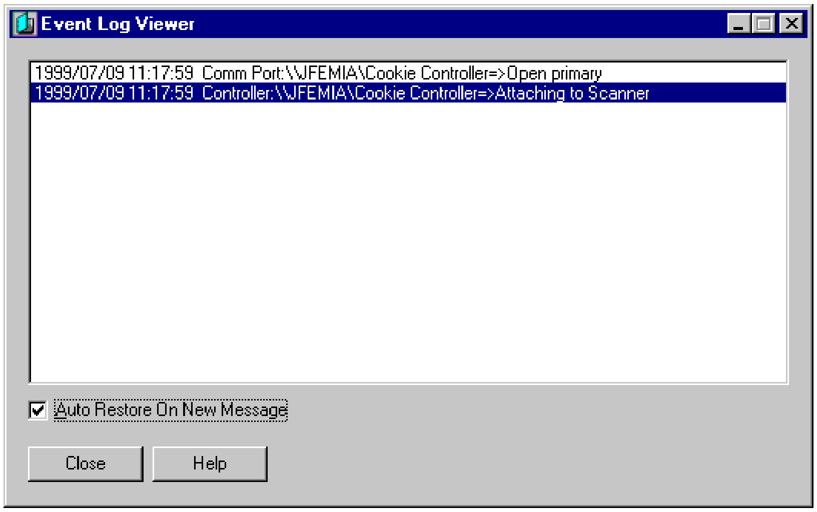
\\JFEMIA\Cookie Controller
Cookie Controller

13. Make sure the new controller name is highlighted, and click OK to close the Select Controller dialog box. Click OK again to close the Controller Properties dialog box and the Controllers dialog box.

14. Choose File > Save Project and Load Runtime.

Running OptoDisplay

OptoDisplay opens and the Event Log Viewer window appears:



```
1999/07/09 11:17:59  Comm Port:\\JFEMIA\Cookie Controller=>Open primary
1999/07/09 11:17:59  Controller:\\JFEMIA\Cookie Controller=>Attaching to Scanner
```

☑ Auto Restore On New Message

In this window you can see that OptoDisplay has connected to the controller through OptoServer. Congratulations! The connection is complete.

Communicating with a DDE-Aware Client

In this second Quick Start section, we will set up OptoServer to provide information to a DDE-aware client, in this case Microsoft Access. In a typical setup, OptoServer and Access would be running on different computers attached to the same network. For this example, however, we'll run them on the same computer to show the configuration process. Briefly, we'll follow these steps:

- Use OptoControl to download a strategy to the controller and verify communication.
- Use the OptoServer Administrator to configure a controller attached to the computer running OptoServer.
- Start OptoServer so that it is ready to be accessed by Access.
- Start Access.

Downloading a Strategy to the Controller and Verifying Communication

1. Start OptoControl by clicking Start and selecting Programs > Opto 22 > FactoryFloor 4.0 > OptoControl > OptoControl.
2. From the File menu, choose Open Strategy. Navigate to Opto 22\FactoryFloor\OptoSrvr\Examples\OptoCtrl and open the sample strategy dde_test.cdb.
3. Configure the strategy for your controller's connection, address, and port settings, and enter "mistic1" as the controller's name.

If you need help, see "Configuring Controllers" in the *OptoControl User's Guide*.

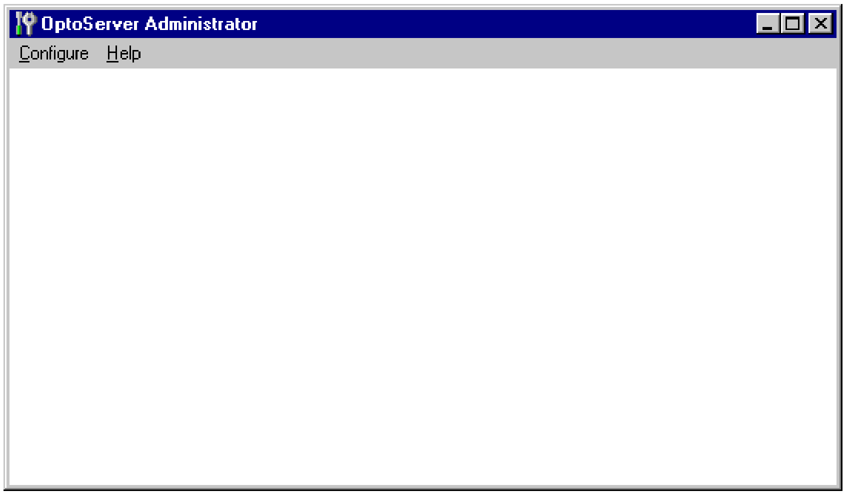
4. Compile the program by selecting Compile > Compile All.
5. Change to Debug mode and run the program.
6. Verify that the controller is communicating with your PC. Observe whether the controller's host port LEDs (ARCNET, COM0, etc.) flash in response to messages from the PC.

   If your controller is responding to your PC with no errors, you're ready to move on to the next step. If you experience problems getting your controller to respond to the OptoControl strategy, make sure the configuration setup and cable wiring between the controller and PC are correct. Consult the *OptoControl User's Guide* and your controller manual to help you troubleshoot the problem.
7. When the controller is communicating with the PC, leave the strategy running and exit OptoControl.

Using OptoServer Administrator to Configure a Controller

1. Start OptoServer Administrator by clicking the Start button and selecting Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer Administrator.

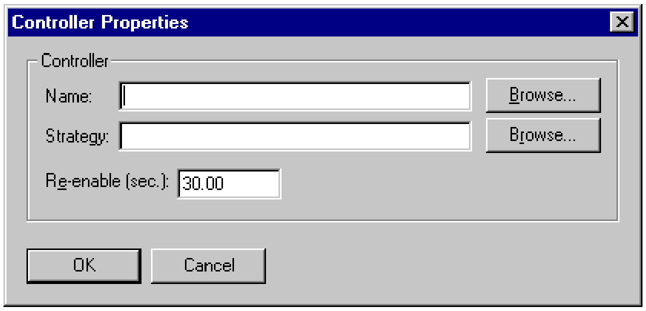
   The OptoServer Administrator main window appears. It is empty:



2. Select Configure > Controllers.

   When you configure a controller for the first time in the OptoServer Administrator, a message says that a file can't be found. Click OK to continue. The necessary file is created.
3. In the Controllers dialog box, click the Add button so you can add a controller.

   The Controller Properties dialog box appears.

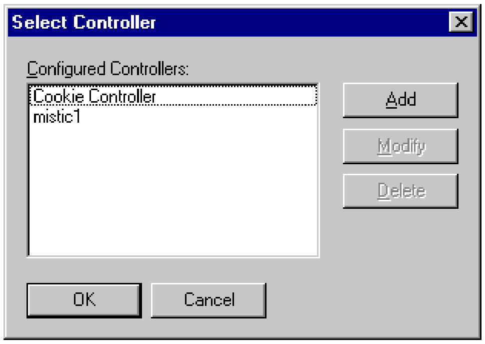


Name:

Strategy:

Re-enable (sec.): 30.00

4. Click the Browse button next to Name.

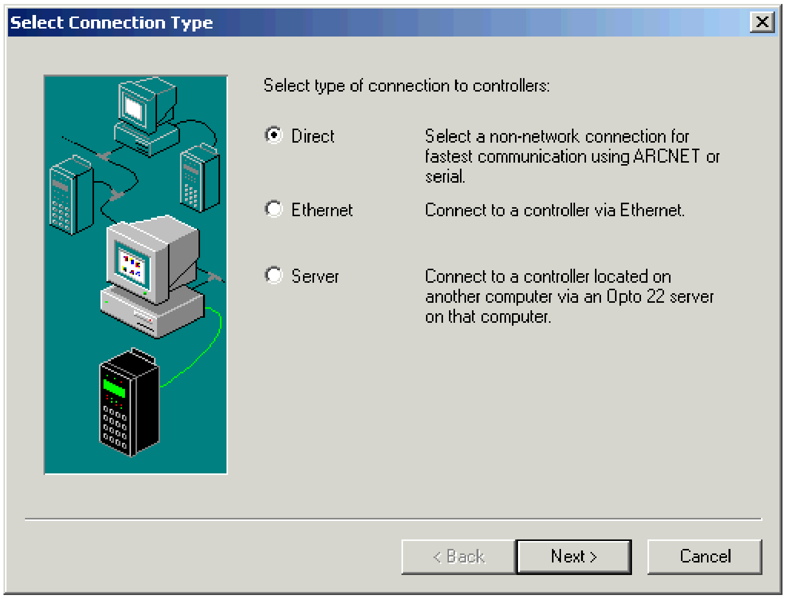


Configured Controllers:

Cookie Controller
mistic1

5. In the Select Controller dialog box, click the mistic1 controller you set up in the OptoControl strategy. Click Modify to check the controller's configuration.

The type of connection should be shown as Direct:



Select type of connection to controllers:

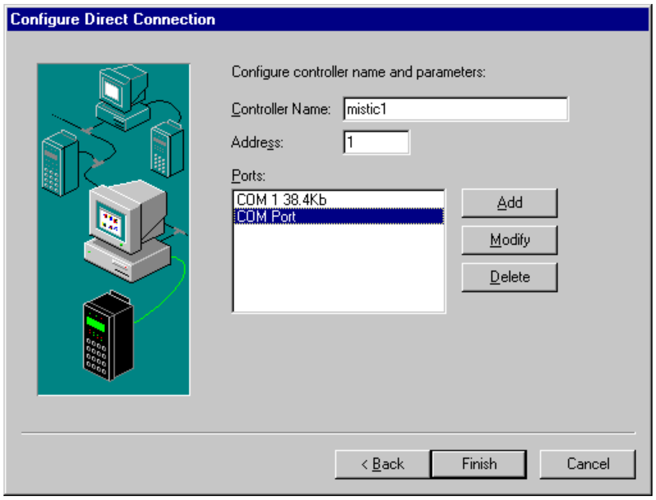
● Direct — Select a non-network connection for fastest communication using ARCNET or serial.

○ Ethernet — Connect to a controller via Ethernet.

○ Server — Connect to a controller located on another computer via an Opto 22 server on that computer.

6. Click Next.

The Configure Direct Connection dialog box appears, with the mistic1 controller listed in the Controller Name field:



Configure Direct Connection

Configure controller name and parameters:

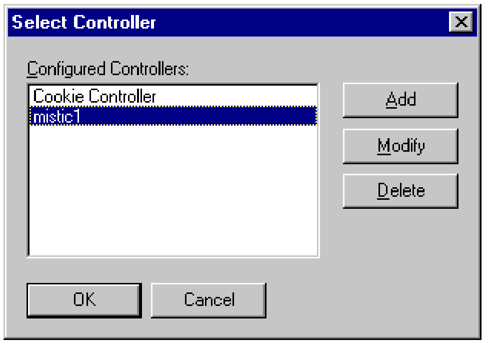
Controller Name: mistic1

Address: 1

Ports:
COM 1 38.4Kb
COM Port

7. If necessary, change the controller's address in the Address field. Highlight the port for direct connection to your controller. If you need to add a port, click Add. (For help, see "Configuring Controllers" in the *OptoControl User's Guide*.) When the address and port are correct, click Finish.

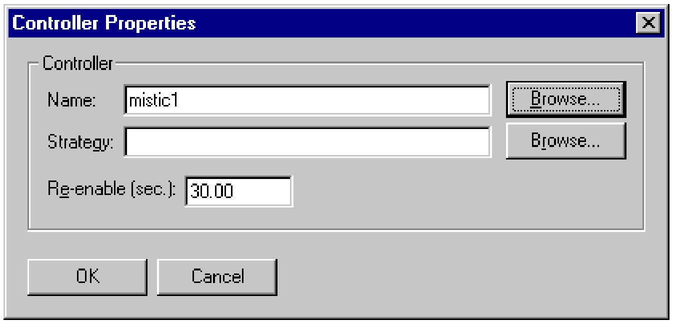
   The newly configured controller appears in the Select Controller dialog box:



Select Controller

Configured Controllers:
Cookie Controller
mistic1

8. Click the new controller to highlight it, and click OK.



Controller Properties

Controller

Name: mistic1

Strategy:

Re-enable (sec.): 30.00

NOTE: If you need to change the Re-enable setting, you can do so here. For more information, see "Understanding Timeout, Retries, and Re-enable Settings" on page 54.

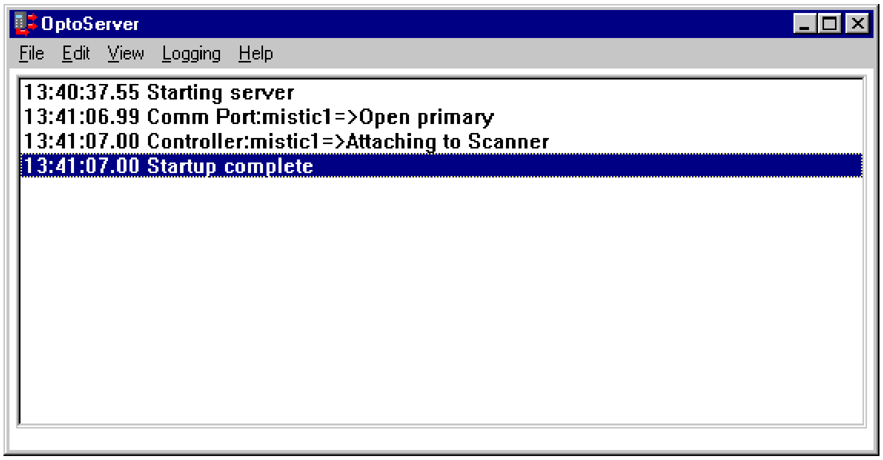
9. Click OK to close the Controller Properties dialog box. Click OK again to close the Controllers dialog box.

10. Exit OptoServer Administrator.

Starting OptoServer

1. Click the Start button and select Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer.

The OptoServer main window appears:



```
13:40:37.55 Starting server
13:41:06.99 Comm Port:mistic1=>Open primary
13:41:07.00 Controller:mistic1=>Attaching to Scanner
13:41:07.00 Startup complete
```

The previous figure shows OptoServer recognizing only the mistic1 controller. Your screen may differ; it shows any controllers you have configured in OptoServer Administrator.

2. Click the Logging menu and make sure the following logging options are checked:

– Log to Window—The main window displays messages.

– Log level Normal—Only significant OptoServer communication events are recorded.

– Log with Time—Messages are saved with a time stamp.

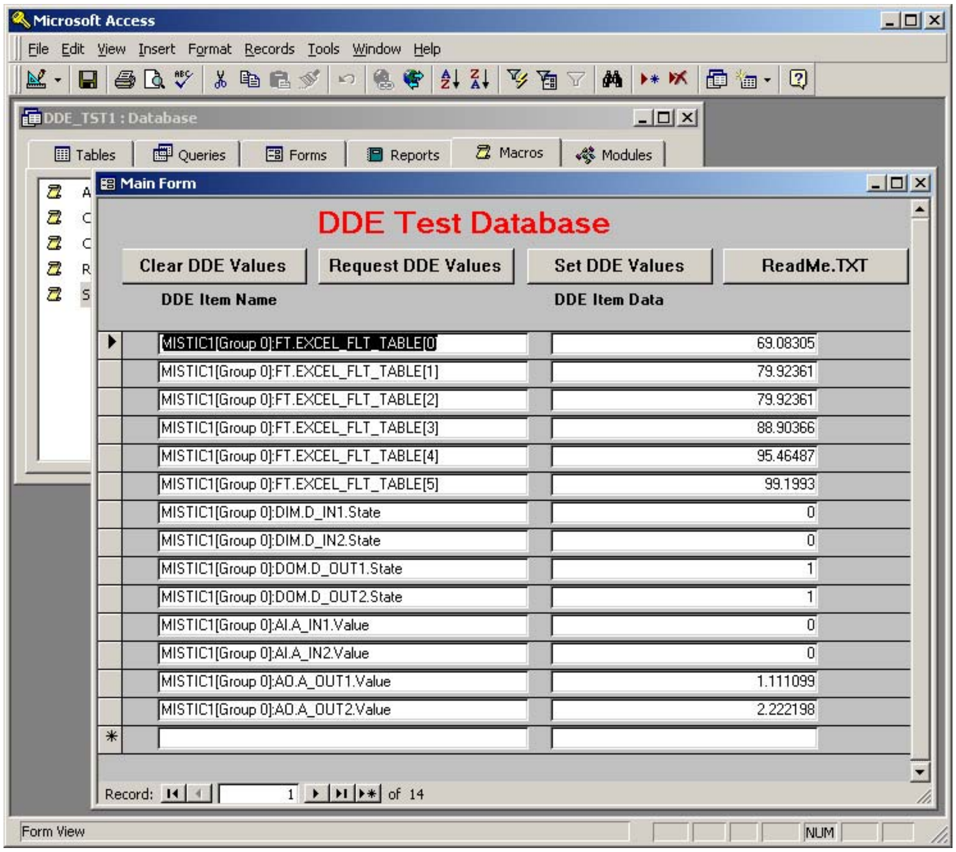
3. Click the minimize button to reduce the OptoServer window to an icon. **Do not exit OptoServer.**

Starting Microsoft Access

1. Start Microsoft Access and open the file OptoSrvr\Examples\Access97\Dde_tst1.mdb.

Depending on the version of Access you are using, you may see warning messages as the file opens. This sample database has been tested with Access 97 only; if you use Microsoft Access 2000 or a later version, you may need to convert or modify the database file and the corresponding Visual Basic for Applications (VBA) macros it uses.

When the file opens, it looks something like this:



DDE Item Name	DDE Item Data
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[0]	69.08305
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[1]	79.92361
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[2]	79.92361
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[3]	88.90366
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[4]	95.46487
MISTIC1[Group 0]:FT.EXCEL_FLT_TABLE[5]	99.1993
MISTIC1[Group 0]:DIM.D_IN1.State	0
MISTIC1[Group 0]:DIM.D_IN2.State	0
MISTIC1[Group 0]:DOM.D_OUT1.State	1
MISTIC1[Group 0]:DOM.D_OUT2.State	1
MISTIC1[Group 0]:AI.A_IN1.Value	0
MISTIC1[Group 0]:AI.A_IN2.Value	0
MISTIC1[Group 0]:AO.A_OUT1.Value	1.111099
MISTIC1[Group 0]:AO.A_OUT2.Value	2.222198

2. Click Request DDE Values and watch as the values in the database fields are updated. Wait a few moments and click again to see how the values reported from the controller change over time.

Congratulations! The DDE setup is complete. For more information about DDE-aware clients, see Chapter 3, "OptoServer and DDE-Aware Clients."

What Is OptoServer?

Introduction

In the previous chapter, we got right to work with OptoServer without discussing much of what it is. In this chapter, we'll define OptoServer and see how you'll benefit by using it with your networked Opto 22 control system.

In This Chapter

About OptoServer

OptoServer is an application that allows its clients to easily access data from networked Opto 22 controllers. Examples of clients are:

- OptoDisplay, OptoControl, and other Opto 22 FactoryFloor applications
- Any OPC 1.0-compliant application, such as a human-machine interface (HMI)
- Any DDE-aware application, such as Microsoft Word
- Any application you have developed that uses the OptoControl SDK.

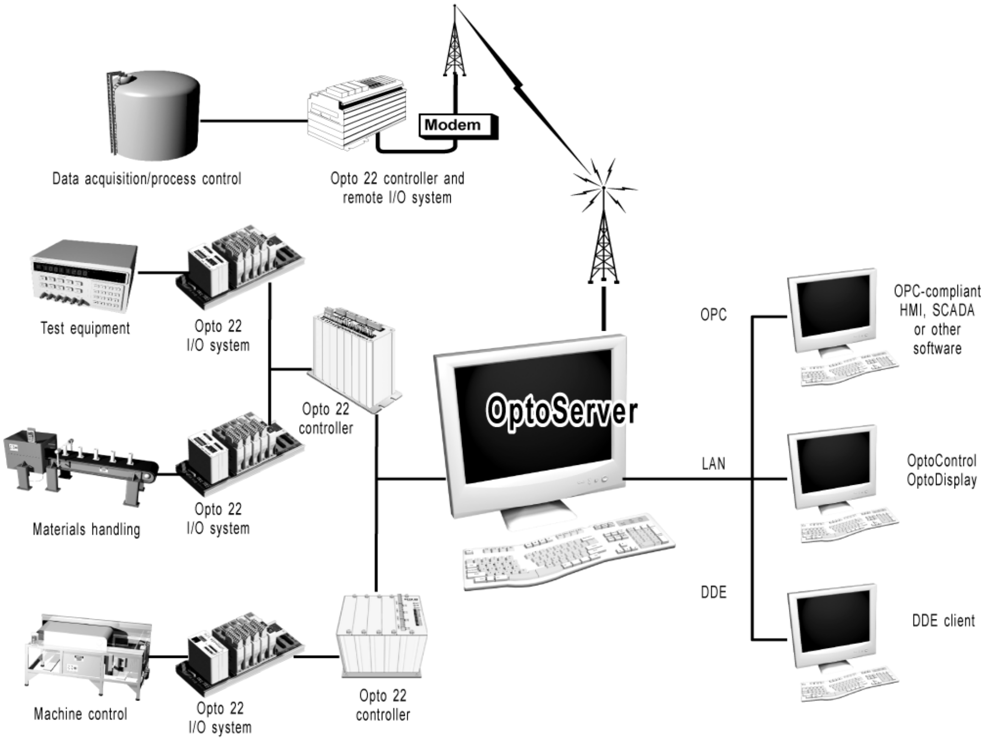
Here are examples of OptoServer in use:

- A computer running OptoDisplay updates data from a controller connected to another computer on the same Microsoft network.
- A computer running OptoControl downloads and debugs a strategy on a controller connected to another computer on the same network.
- An OPC-compliant or DDE-aware application uses data from an Opto 22 controller that is either on the same computer or on another computer accessible via NetDDE.

Why Use OptoServer?

OptoServer's advantage is that it provides real-time information to multiple PCs needing data from the same Opto 22 controller(s), without slowing down the controller's throughput.

The following figure shows a sample OptoServer setup.



As you can see, several computer terminals, each running Opto 22, OPC-compliant, or DDE-aware clients, are networked to a PC running OptoServer. The PC running OptoServer has a direct physical connection to Opto 22 controllers via ARCNET, RS-485, Ethernet, or any other Opto 22 controller-supported communication option.

For OPC and DDE clients, OptoServer scans controllers at the rate you set and sends data to clients. For Opto 22 clients such as OptoDisplay, OptoServer provides a connection to remote controllers and a cache option to reduce the amount of scanning necessary. Without OptoServer, each client would scan the Opto 22 controllers independently for data updates—and probably slow down the controllers' throughput rates.

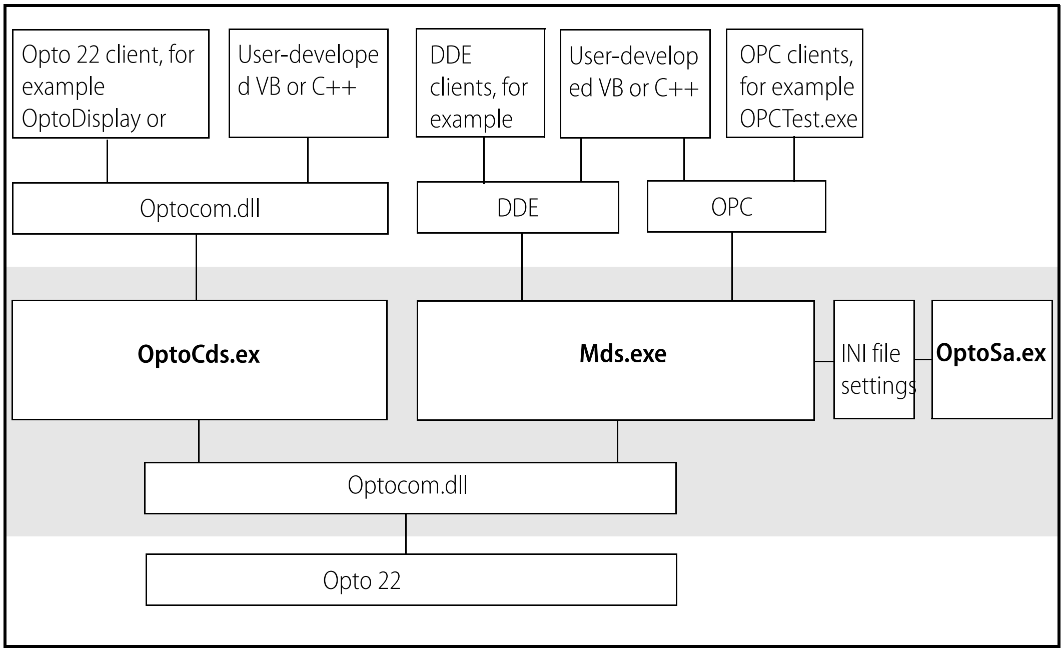
The OptoServer Package

Three programs make up the OptoServer package:

- **OptoServer Administrator (OptoSa.exe)**—Used to select the controllers that OPC, DDE, and OptoDisplay clients need to access. Also used to set scan times for OPC and DDE clients or to configure the behavior of DDE transactions. (Scan times for OptoDisplay are configured in the OptoDisplay Configurator.)

- **OptoServer (Mds.exe)**—A server that scans for data requested by OPC or DDE clients and sends data from client(s) to controller(s).
- **Communication Data Server (OptoCds.exe)**—A server that communicates with other Opto 22 applications such as OptoControl and OptoDisplay (or any other application that uses OptoCom.dll). OptoCds starts automatically when OptoServer starts or can be launched independently.

The following figure illustrates how these programs communicate with each other and their clients. All the applications in the shaded area are on one computer; the applications above the shaded area can be on the same computer or on multiple computers that are networked.


Opto 22 client, for example OptoDisplay or
User-develope d VB or C++
DDE clients, for example
User-develop ed VB or C++
OPC clients, for example OPCTest.exe
Optocom.dll
DDE
OPC
OptoCds.ex
Mds.exe
INI file settings
OptoSa.ex
Optocom.dll
Opto 22

Client/Server Basics

OptoDisplay projects, OptoControl strategies, and OPC-compliant or DDE-aware applications running under Microsoft Windows 95, Windows 98, or Windows NT can exchange data on a real-time basis by engaging in a conversation—a series of exchanged messages between applications to request and supply data. The application initiating the conversation and requesting or sending (”poking”) data is the client. The application supplying the data is the server.

OptoServer is an example of a server, while OptoDisplay, an OPC-compliant HMI, and Microsoft Word are examples of clients. Clients and servers are typically connected together over a local area network such as Ethernet or ARCNET. Clients must be registered with the server to receive or send information.

OPC and DDE Clients

OptoServer (Mds.exe) scans the controllers on the network and provides data to OPC and DDE clients. Scan rates for OPC-compliant and DDE-aware clients are set up in the OptoServer Administrator program. See "Optimizing OptoServer Scanning" on page 19 for more information.

OPC-compliant and DDE-aware clients are sent information on a report-by-exception basis. This means they are only sent data that is newer than or different from the last information OptoServer sent.

Opto 22 Clients

The Communication Data Server (OptoCds.exe) provides a conduit for OptoDisplay and other Opto 22 clients to scan controllers on the network. Because these clients do their own scanning, scan rates are set in the client. For example, you can set scan rates for OptoDisplay in the OptoDisplay Configurator.

Since the bottleneck in communications is at the controller, OptoCds uses a cache to reduce unnecessary or repetitive scanning. OptoCds stores the scanned data from controllers in the cache. If a client such as OptoDisplay requests data that is in the cache, the server delivers it immediately. This cache is similar to a disk cache or Internet cache. It improves throughput because it's much faster to retrieve data from the cache in memory than from the controller. Here's an example of how the cache works:

1. OptoDisplay Client A requests data for a window.
2. The request goes through the Communication Data Server to the controller.
3. The controller responds through OptoCds to OptoDisplay Client A. OptoCds stores the response, as well as the corresponding command and the identity of the controller, in the cache.
4. OptoDisplay Client B requests the same data for the same window.
5. The request goes through OptoCds, which checks the cache. If the data is still fresh, OptoCds delivers the previous response stored in the cache to OptoDisplay Client B. If the data is stale, OptoCds contacts the controller, delivers the new data to the client, and stores the new data in the cache.

You can turn caching on or off in the Communication Data Server and also set the freshness time for the data. See "Working with Communication Data Server" on page 26 for more information.

Using OptoServer Efficiently

If the computer running OptoDisplay has a connection to the controller, you should run OptoDisplay using that connection rather than using OptoServer. Using OptoServer between a controller and OptoDisplay is slower and less efficient than a connection without OptoServer. One of the main purposes of OptoServer is to provide a connection between computer and controller when no other connection exists.

You may run into a bottleneck if OptoServer is run on only one computer and is used for a large number of clients. A bottleneck can occur for two reasons: first, the volume of data may be too much for one OptoServer application and computer to handle, and second, OptoServer scans multiple controllers sequentially. In contrast, if several computers are running OptoDisplay projects, they scan the controllers essentially in parallel; that is, each project scans the controllers independently of the other projects. If a bottleneck occurs, run OptoServer on more than one computer and divide the clients evenly among them.

Optimizing OptoServer Scanning

For OPC and DDE clients, OptoServer scans controllers for data at rates you can set. Up to seven scan rates can be configured, so you can have OptoServer scan data at different rates depending upon the importance or nature of the data. For example, digital points can be scanned every second, while a group of temperature inputs can be scanned every five seconds, since temperatures usually change slowly.

If a PC cannot keep up with the scanning requirements, you'll receive "scanner overrun" notifications. These notifications are not error messages; they occur when the time it takes for the controller to scan the data is longer than the configured scan time. OptoServer gets a "server data" notification if the server scans for new data before it can send already-scanned data to the client.

NOTE: For OptoDisplay clients, scanner overrun notifications appear in OptoDisplay. You can enable or disable notification in the Refresh Times dialog box in the OptoDisplay Configurator.

Scanning and OptoDisplay

When OptoDisplay scans for data, it combines data items into groups. Due to communications overhead, sending data in one large group is faster than sending several smaller data requests.

Keep in mind the following data group characteristics and how they affect scanning:

- Each OptoDisplay draw window has its own group of data items.
- Data items with different scan times from the same OptoDisplay draw window are combined into separate groups.
- The maximum number of data items in a group is 50.
- Strings are not combined into groups.
- Data written to a controller is not grouped.

Adjusting Scan Times

Here's a suggested method you can use to find an optimum scan time for your OptoServer system and help avoid scanner overrun errors. The goal is to scan the controller(s) as often as possible to provide the latest data to clients, without affecting the controller's throughput and giving constant scanner overruns.

Follow these steps to adjust scan times:

1. Set all client scan times to a long interval to ensure a "quiet time" (no communication activity) between scans. Start out with a 30-second interval.
   - For OptoDisplay clients, set scan times in the OptoDisplay Configurator. (See the *OptoDisplay User's Guide* for instructions.)
   - For DDE-aware clients, set scan times in the OptoServer Administrator as explained in "Setting Refresh Times" on page 22.
2. Run your OptoControl, OptoServer, and OptoDisplay applications.

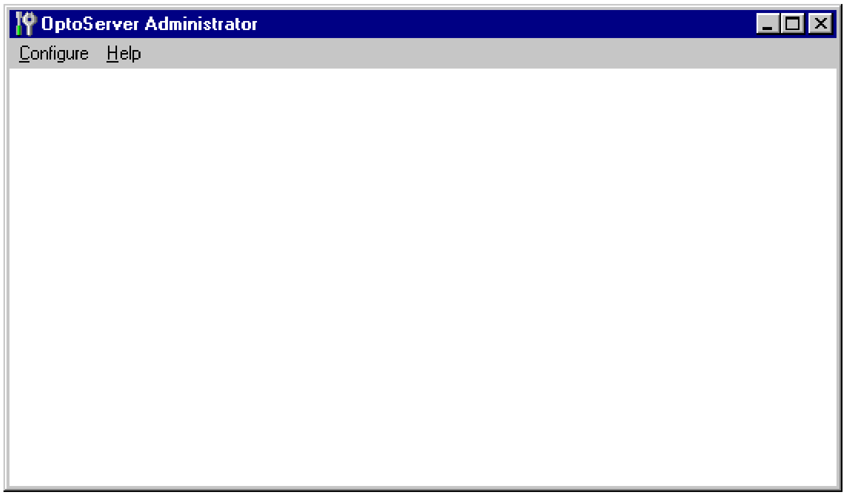
3. Use a stop watch to time the duration of communication activity during a scan interval. Use the communication LEDs on the Opto 22 controller to monitor the activity.
4. Adjust your scan times to a value greater than the communication activity duration you measured.
5. Use prime numbers for scan values. That way scan values won't be multiples of each other and occur at the same time.
6. To further adjust your scan times:
   a. Shorten the digital I/O scan times and lengthen the analog I/O scan times.
   b. Set the refresh group name's freshness value equal to its scan time. By doing so, the controller won't have to scan I/O units as frequently.

Working in OptoServer Administrator

You use OptoServer Administrator to configure controllers and to set refresh times for OPC-compliant and DDE-aware clients.

To open OptoServer Administrator, click the Start button and choose Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer Administrator.

The OptoServer Administrator main window appears:



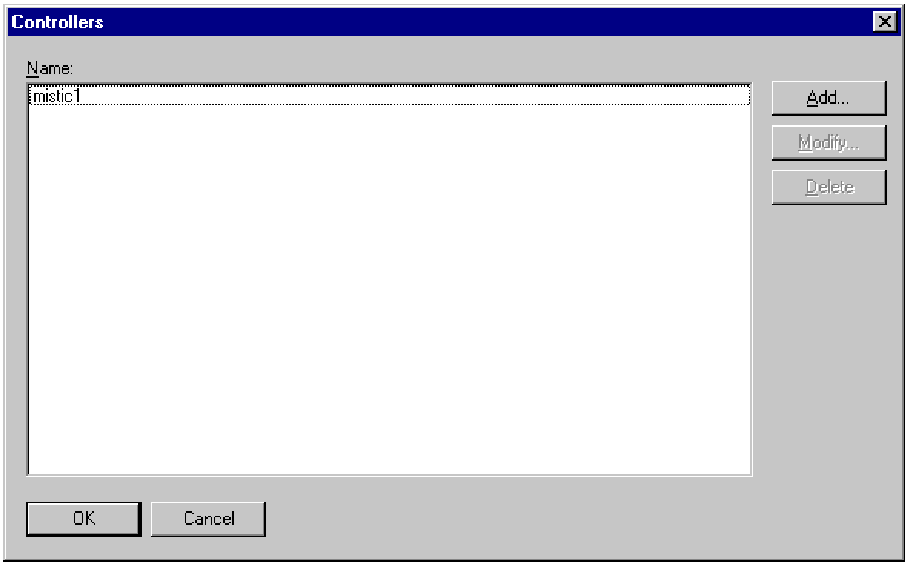
Online Help is available in OptoServer Administrator from the Help menu.

Configuring Controllers

1. In the OptoServer main window, select Configure > Controllers.
2. If a message says that the OptoMds.ini file cannot be found, click OK to continue.

   The necessary file is created.

   The Controllers dialog box lists all currently configured controllers:



mistic1

Adding, changing, and deleting controllers is the same as in OptoControl. For detailed steps, see “Working With Controllers” in the *OptoControl User’s Guide*.

Setting Refresh Times

You use OptoServer Administrator to set scan rates and freshness values for OPC-compliant and DDE-aware clients, such as Microsoft Word. Refresh names are used in client messages. (See Chapters 3 and 4 for more information on client messages.) Here is a sample DDE message using the default refresh name “Group 0”:

```
MDS|OPTO_MDS_0!’MyMistic[Group 0]:I.TEST_INT’
```

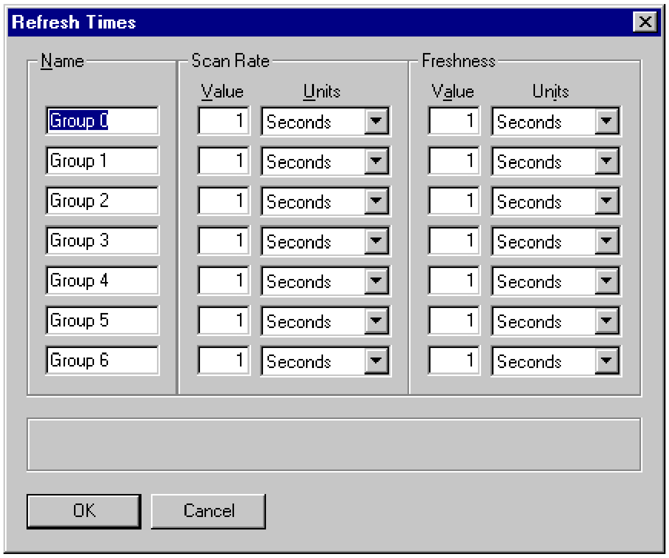
You can set up to seven refresh names, each with its own refresh time, and a client can use more than one refresh name. Using more than one name is helpful if you have groups of I/O with different scan rates, for example, analog I/O points to read the outdoor temperature and digital I/O sensing whether a security gate is open. Since outdoor temperatures are fairly stable, the scan rate for the analog I/O would typically be longer than the scan rate for the digital I/O group.

NOTE: See “Optimizing OptoServer Scanning” on page 19 for additional information about scanning rates and refresh times, especially if you are using OptoServer with OptoDisplay as well as DDE-aware clients.

Follow these steps to set scan rates and refresh times.

1. In the OptoServer Administrator main window, choose Configure > Refresh Times.

The Refresh Times dialog box appears.



Name	Scan Rate		Freshness	
	Value	Units	Value	Units
Group 0	1	Seconds	1	Seconds
Group 1	1	Seconds	1	Seconds
Group 2	1	Seconds	1	Seconds
Group 3	1	Seconds	1	Seconds
Group 4	1	Seconds	1	Seconds
Group 5	1	Seconds	1	Seconds
Group 6	1	Seconds	1	Seconds

2. In the Name column, type a name for the first refresh time you want to set up, or use the default name.

The name can be up to 15 characters long. Do not use the ! and | characters in the name. Spaces are valid characters, but remember to include them when creating your client's DDE or OPC message. Don't omit them or substitute an underscore (_) for a space.

3. In the Scan Rate columns, specify how often OptoServer scans the controllers for data by entering a value from zero to 9,999 in the Value field. Select a unit of time from the Units drop-down list (milliseconds, seconds, minutes, hours, days, or months).

The scan rate must be greater than or equal to the value in the Freshness field. A higher scan rate means more time between I/O readings and fewer times the controller is scanned.

4. In the Freshness field, specify how current (fresh) the data from the controller must be by entering a value from zero to 9,999 in the Value field. Select a unit of time from the Units drop-down list.

The freshness value must be less than or equal to the value in the Scan Rate field. If the data available from the controller is older than the time specified in the Freshness field, the controller scans the I/O units for new data.

5. Repeat from step 2 for each refresh time you want to set up.

6. When you have finished setting refresh times, click OK.

You return to the OptoServer Administrator main window.

Changing the Poke Order

If you are using DDE-aware clients and "poking" (writing) multiple values to integer or float tables in OptoControl, you can change the order in which the data is poked.

By default, data is poked in reverse order. For example, in a five-element in a Microsoft Word document, element 1 in the table maps to element 5 in the OptoControl table, element 2 maps to element 4, and so on. To have elements match 1 to 1, 2 to 2, and so on, follow these steps:

1. In OptoServer Administrator, open the Configure menu.

2. Click Reverse Table Poke Order to uncheck it.

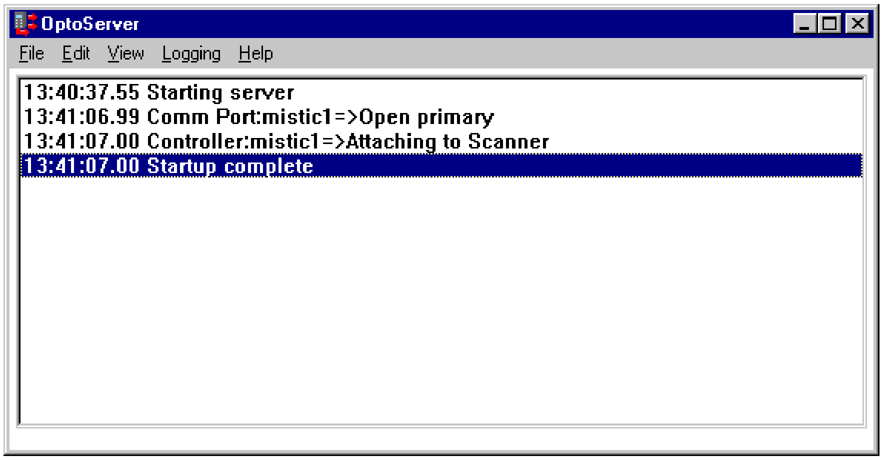
   Pokes will no longer be in reverse order.

Working with OptoServer (Mds.exe)

OptoServer (Mds.exe) is used to communicate with OPC or DDE clients. Besides scanning data for these clients, OptoServer also provides a way to see the number of attached clients and log messages to the screen or a file. Before starting OptoServer, you should have already configured a controller in OptoServer Administrator. (See "Configuring Controllers" on page 21.)

To open OptoServer (Mds.exe), click the Start button and select Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer.

The OptoServer main window appears. The following figure shows OptoServer recognizing a mistic1 controller. Your screen shows any controllers you have configured in OptoServer Administrator.



```
13:40:37.55 Starting server
13:41:06.99 Comm Port:mistic1=>Open primary
13:41:07.00 Controller:mistic1=>Attaching to Scanner
13:41:07.00 Startup complete
```

If the OptoServer main window is blank, do the following:

- Configure a controller in OptoServer Administrator. (See page 21.)
- Click the Logging menu and make sure Log to Window is checked.

Online Help is available in OptoServer from the Help menu. Note that Communication Data Server (OptoCds.exe) opens when OptoServer opens. For information on using Communication Data Server, see page 26.

Viewing the Number of Clients

To view the number of DDE and OPC clients that are connected to OptoServer, choose View > Clients.

A small window appears, showing the number of clients connected to OptoServer. The number includes DDE and OPC clients only. Clients served through the Communication Data Server—for example, OptoDisplay—are not included.

Logging Messages

Status and error messages are normally logged to the OptoServer main window so you can monitor communication with DDE and OPC clients. You can also log messages to a log file. For a list of messages, see page 56.

Logging to the OptoServer Window

1. To start or stop logging messages to the main window, choose Logging > Log to Window.

   A check mark next to the menu item starts logging; removing the check mark stops logging.

2. To time or date stamp messages, choose one of the following:
   - Logging > Log with Date to include the date of each message in the format DD-MM-YYYY (D=day, M=month, Y=year).
   - Logging > Log with Time to include the time of each message in the format HH:MM:SS:TT. (H=hours, M=minutes, S=seconds, T=hundredths of a second. Time is based on a 24-hour clock.)
3. To select a different level of messages to send to the main window, choose one of the following:
   - Logging > Log Level Normal to show errors and client connections or terminations.
   - Logging > Log Level Diagnostics to show normal logging messages plus all data sent from OptoServer to clients.
   - Logging > Log Level Advanced to show normal and diagnostics data plus all internal OptoServer events. Internal OptoServer events can be used by Opto 22 Product Support staff to help troubleshoot problems if needed.
4. To save your message settings, choose one of the following:
   - File > Save Settings Now to save message log settings immediately.
   - File > Save Settings on Exit to save message log settings when you exit OptoServer.
5. To delete all messages currently in the main window, choose Logging > Clear Log Window.

   If you are logging to a file (see the next section), the log file is not cleared.

6. To see all of a long message that is truncated in the main window, double-click the message or highlight it and choose Logging > View Log Entry.
7. To copy a message to the Windows clipboard, so you can paste it into another Windows application, highlight the message in the OptoServer main window and choose Edit > Copy.

Logging to a File

You can log messages both to the window and to a file so they can be edited, printed, and saved. The file's default file name is mdslog.log, and it is created in the OptoServer root directory (typically Opto 22\FactoryFloor\OptoSrvr).

1. (Optional) If you want to change the name or location of the log file from the default, choose Logging > Select Log File. Navigate to the location where you want the file to be, type in the file name, and click OK.
2. In the OptoServer main window, choose Logging > Log to File so that the menu item is checked.

   The messages begin appearing in the log file.

3. To select a different level of messages to send to the log file, choose one of the following:
   - Logging > Log Level Normal to show errors and client connections or terminations.
   - Logging > Log Level Diagnostics to show normal logging messages plus all data sent from OptoServer to clients.
   - Logging > Log Level Advanced to show normal and diagnostics data plus all internal OptoServer events. Internal OptoServer events can be used by Opto 22 Product Support staff to help troubleshoot problems, if needed.
4. To save your message settings, choose one of the following:
   - File > Save Settings Now to save message log settings immediately.
   - File > Save Settings on Exit to save message log settings when you exit OptoServer.
5. To edit the log file, choose Logging > Edit Log File.

   The Microsoft Windows Notepad editor starts and opens the message log file for editing.

Exiting OptoServer

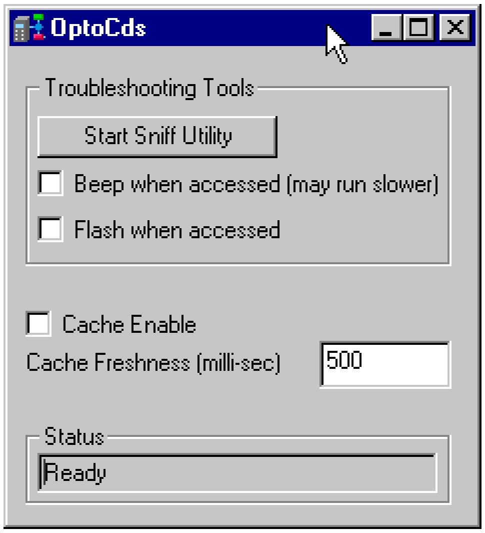
When you exit OptoServer, OPC clients generally stop registering values when the server is disconnected and may indicate a failure to read. DDE clients, however, do not automatically detect the server's absence.

Working with Communication Data Server

Communication Data Server (OptoCds.exe) communicates with other Opto 22 applications such as OptoControl and OptoDisplay. Communication Data Server opens when you launch OptoServer, or you can launch OptoCds separately.

1. To launch OptoCds separately, click the Start button and select Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > Communication Data Server.

   The Communication Data Server window appears:



□ Beep when accessed (may run slower)

□ Flash when accessed

□ Cache Enable

Cache Freshness (milli-sec) 500

Status: Ready

2. To turn on the Communication Data Server's caching feature, click Cache Enable to place a check mark in the box.

   See "Opto 22 Clients" on page 18 for more information about the cache.

3. If Cache Enable is checked, in the Cache Freshness field enter the length of time in milliseconds that data in the server's cache should be considered fresh.

   If data has been cached longer than this, OptoServer does not deliver the data, and the client has to scan the controller.

Using Troubleshooting Features in Communication Data Server

Three troubleshooting tools are available from the OptoCds window.

To launch OptoSniff, a handy utility for monitoring communication between a PC and a controller, click the Sniff Utility button. For more information on OptoSniff, see the Troubleshooting appendix in the *OptoControl User's Guide*.

To sound a beep whenever a client requests data, check Beep when accessed. Normally this box should be unchecked, because beeps slow performance dramatically. When you're troubleshooting, however, the beeps verify that OptoCds is being accessed.

To have the OptoCds window's title bar flash whenever a client requests data, check Flash when accessed. If the window is minimized, it will flash in the taskbar at the bottom of the computer screen.

OptoServer and DDE-Aware Clients

Introduction

Part of the beauty of OptoServer is its ability to provide information from Opto 22 controllers to third-party applications that can communicate using Dynamic Data Exchange (DDE). This chapter discusses general concepts that apply to all DDE-aware applications and includes a brief procedure for getting Microsoft Visual Basic programs working with OptoServer. Sample programs used to demonstrate this procedures are included with OptoServer.

In This Chapter

General Concepts

DDE-aware applications are updated in real time by using OptoServer to scan Opto 22 controllers for information. For example, if you open an OptoServer-registered Microsoft Word document, DDE-linked fields are automatically updated as their corresponding Opto 22 controller data values change. You can do the same thing with Microsoft Visual Basic or any other program that supports DDE.

Each DDE-aware application has its own set of syntactical rules to create the DDE link messages that connect it to OptoServer. This section discusses the common elements of DDE messages. Be sure to consult your DDE-aware program manuals for individual differences in DDE linking.

DDE Messages

A DDE-aware client sends a DDE message to initiate a conversation with OptoServer and to request data. Although each application has its own unique syntax to develop messages, they all consist of three common character expressions: Application, Topic, and Item.

Application

The application is simply the name of the server supplying the data information. OptoServer uses MDS for its name. Actual string expressions depend on your DDE-aware application's syntax and what node OptoServer is running on relative to the DDE-aware application.

Topic

A topic expression identifies a category of data from the application. For OptoServer, you should generally use the topic expression OPTO_MDS_0. Topic names are not case-sensitive, but note that they contain underscores rather than spaces. Remember to include underscores in your topic expressions.

If you are working across a network, the topic has the syntax: <topic>$, so the topic would be OPTO_MDS_0$.

NOTE: OptoServer also creates topics based upon the controller information you have set up. These topics will be in the following format:

OptoCtrl.ControllerName

where ControllerName is the name you supplied for the controller when it was configured.

Item

The item expression identifies a data item from an OptoControl strategy. The format for an OptoControl DDE item has the following syntax in all applications:

ControllerName[RefreshGroupName]:ItemType.ItemName[start-end].Field.BITn

Each parameter is described in the Item Parameters table on the following page.

ControllerName is normally included. However, if you need to swap controllers, for example to shift to a backup controller, you can leave the controller name out of the item expression. If it is left out of the item expression, OptoServer looks for the controller name as part of the topic expression.

RefreshGroupName, ItemType, and **ItemName** must always be included.

The **[start-end], Field,** and **BITn** parameters are applicable to some data types only. The Item Types table starting on page 32 describes item types in detail.

NOTE: The ItemName must be defined in all capital letters in the OptoControl strategy.

Item Parameters

Parameter	Description
ControllerName	The name of the controller to scan for data. This parameter is configured in the OptoServer Administrator with the Configure Controllers command. This parameter is normally included in the item expression, but it does not have to be. If it is not included in the item expression, OptoServer looks for it as part of the topic expression.
[RefreshGroupName]	The scan rate group for a data item. This parameter is required when a DDE-aware client requests data from OptoServer. This parameter is configured in OptoServer Administrator. (See page 22.)
ItemType	Defines the data type of the item requested. Valid ItemTypes are defined in the Item Types table on page 32. The identifier used in the message is listed in the ItemType column.
ItemName	The actual data item's name in the OptoControl strategy.
[start-end]	Applies only to data items of type float table or integer table, and indicates the range of elements for ItemNames of these types. "Start" is the first element in the range and "end" is the last element in the range. Single table elements can be requested by specifying only a start parameter. Up to 50 elements can be specified in one request, so you must limit the start-end range to request only 50 elements (0–49, 101–150, and so on). Only one element from a string table may be requested at a time. See "Changing the Poke Order" on page 23 for information on the order in which elements are poked to tables.
Field	Applies only to data items with more than one value associated. For example, a PID loop item type has a setpoint, a gain, a derivative, and many other related fields. Refer to the "Field" column in the Item Types table for an item type's applicable fields.
BITn	Applies only to items of type integer and integer table elements. It is the literal character string "BIT" followed by a number between zero and 31, inclusive (for example, BIT28).

Item Types

The following table shows supported item types and fields for the Item expression.

NOTE: This table spans several pages.

Description	Item Type	Field	Actual Base Type	Access
32-bit Integer[1]	I	None	integer	R/W
Float	F	None	float	R/W
String	S	None	string	R/W
Timer	T	None	float	R/W
32-bit Integer Table	IT	[start-end][2]	integer table	R/W
		[element]	integer	R/W
Float Table	FT	[start-end][2]	float table	R/W
		[element]	float	R/W
String Table	ST	[element]	string	R/W
Digital Simple I/O Unit (G4D16LS)	DIOS	Enable	discrete	R/W
Digital Multifunction I/O Unit (B100, B3000, B3000-HA, G4D16R, G4D16L)	DIOM	Enable	discrete	R/W
Analog Multifunction I/O Unit (B200, B3000, B3000-HA, G4A8R, G4A8L)	AIO	Enable	discrete	R/W
Digital Remote Simple I/O Unit (SNAP-BRS, G4D32RS)	DIORS	Enable	discrete	R/W
Digital and Analog Mixed I/O Unit (SNAP-B3000-ENET)	MIX	Enable	discrete	R/W
Digital-only Ethernet I/O Unit (SNAP-ENET-D64)	D64	Enable	discrete	R/W
Analog Input Point	AI	Value	float	R
Analog Output Point	AO	Value	float	R/W

[1] *If you are using 64-bit integers in OptoControl, convert them to two 32-bit integers using the commands Get High Bits of Integer 64 and Get Low Bits of Integer 64.*

[2] *Table range is limited to 50 elements.*

[3] *OptoControl chart "state" values are as follows: (-3) Waiting for I/O, (-1) Start, (0) Stop, (1) Suspend.*

Description	Item Type	Field	Actual Base Type	Access
PID Loop	PID	Input	float	R
		Output	float	R/W
		Setpoint	float	R/W
		Gain	float	R/W
		Integral	float	R/W
		Derivative	float	R/W
		Process Variable	float	R/W
		Setpoint Max	discrete	R/W
		Setpoint Min	discrete	R/W
		Input Range	discrete	R/W
		Activate	discrete	R/W
		Auto	discrete	R/W
		Output Enable	discrete	R/W
		Output Track	discrete	R/W
		Host for Process Variable	discrete	R/W
		I/O for Setpoint	discrete	R/W
		Average Input	discrete	R/W
Digital Simple Input Point	DIS	State	discrete	R
Digital Multifunction Input Point	DIM	State	discrete	R
Digital Remote Simple Input Point	DIRS	State	discrete	R
Counter	CNT	State	discrete	R
		Counts	integer	R
Quadrature Counter	QUAD	State	discrete	R
		Counts	integer	R
On Time Totalizer	TOTON	State	discrete	R
		On Time Total	float	R
Off Time Totalizer	TOTOFF	State	discrete	R
		Off Time Total	float	R
On Pulse Measurement	PULON	State	discrete	R
		On Pulse Measure	float	R

[1] *If you are using 64-bit integers in OptoControl, convert them to two 32-bit integers using the commands Get High Bits of Integer 64 and Get Low Bits of Integer 64.*

[2] *Table range is limited to 50 elements.*

[3] *OptoControl chart “state” values are as follows: (-3) Waiting for I/O, (-1) Start, (0) Stop, (1) Suspend.*

Description	Item Type	Field	Actual Base Type	Access
Off Pulse Measurement	PULOFF	State	discrete	R
		Off Pulse Measure	float	R
Frequency Measurement	FREQ	State	discrete	R
		Frequency	integer	R
Period Measurement	PERIOD	State	discrete	R
		Period	float	R
Digital Simple Output Point	DOS	State	discrete	R/W
Digital Multifunction Output Point	DOM	State	discrete	R/W
		On Time Percent	float	R/W
		Period	float	R/W
Digital Remote Simple Output Point	DORS	State	discrete	R/W
Digital TPO Output Point	TPO	State	discrete	R/W
		On Time Percent	float	R/W
		Period	float	R/W
Digital Event/Reaction	ERD	Scan Enable	discrete	R/W
		Has Occurred	discrete	R
		Is Occurring	discrete	R
Analog Event/Reaction	ERA	Scan Enable	discrete	R/W
		Has Occurred	discrete	R
		Is Occurring	discrete	R
Chart	CHART	Run	discrete	R/W
		Stop	discrete	R/W
		Suspend	discrete	R/W
		Continue	discrete	W
		State[3]	integer	R/W

[1] *If you are using 64-bit integers in OptoControl, convert them to two 32-bit integers using the commands Get High Bits of Integer 64 and Get Low Bits of Integer 64.*

[2] *Table range is limited to 50 elements.*

[3] *OptoControl chart "state" values are as follows: (-3) Waiting for I/O, (-1) Start, (0) Stop, (1) Suspend.*

Description	Item Type	Field	Actual Base Type	Access
Controller	CTRLR	Free Memory	integer[1]	R
		Time Stamp	string	R
		Date Stamp	string	R
		Time/Date	string	R
		Filename	string	R
		# of Errors	integer[1]	R
		Current Error	integer[1]	R

[1] If you are using 64-bit integers in OptoControl, convert them to two 32-bit integers using the commands Get High Bits of Integer 64 and Get Low Bits of Integer 64.
[2] Table range is limited to 50 elements.
[3] OptoControl chart "state" values are as follows: (-3) Waiting for I/O, (-1) Start, (0) Stop, (1) Suspend.

NetDDE Messages

OptoServer automatically starts NetDDE on the server PC if it is not already running, and NetDDE remains running when OptoServer is closed. If the client is on a different PC (a different network node), you need to start NetDDE on the client computer.

NetDDE share names are slightly different in 32-bit Windows (Windows 95/98 and Windows NT) than in 16-bit Windows (Windows for Workgroups, for example). Because 32-bit Windows doesn't allow periods in the share name, the shares take the format Opto_Mds_0$ instead of $Opto_Mds_0.Dde. OptoServer creates its shares if they don't already exist. This difference affects any DDE client that uses NetDDE communications to Mds.

DDE and Case Sensitivity

Tagnames on the controller must be uppercase in order to be accessible to DDE clients. For example, a variable called MY_INT can be accessed via DDE, but My_Int cannot. The reason is that although DDE topic and item strings are **not** case-sensitive, OptoControl **is** case-sensitive. Since OptoServer must make an assumption about the case of a tag, it assumes that tags are uppercase. In other words the DDE expression

```
=MDS|OPTO_MDS_0!'MyRtu[Group 0]:I.My_Int'
```

would actually access an integer called MY_INT.

DDE expressions are not case-sensitive, which means that the application, topic, and item may be upper, lower, or mixed case.

Visual Basic and OptoServer

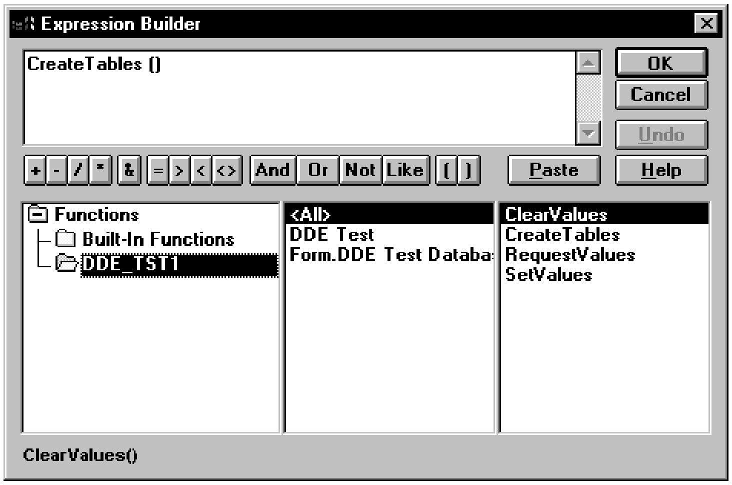
You can also use OptoServer to update your Visual Basic applications with controller information. This section discusses the Visual Basic programming elements used in a sample program provided with OptoServer, which demonstrates how to poke and request controller information into a Visual Basic program by using a timer. The example is written for Visual Basic 5.0.

Go through the next section to open and view the main components of the sample program. For specific details about DDE linking with Visual Basic, consult your Microsoft Visual Basic manual.

Linking Visual Basic and OptoServer

1. Follow the steps in "Downloading a Strategy to the Controller and Verifying Communication" on page 8 to verify that your controller and OptoControl strategy are running properly.
2. Also follow the steps in "Using OptoServer Administrator to Configure a Controller" on page 9, but before closing the OptoServer Administrator, configure a Refresh Group name for use by the DDE-aware client. (See "Setting Refresh Times" on page 22 for an explanation of the dialog box.)
3. Close OptoServer Administrator.
4. Start OptoServer (Mds.exe) and verify that your controller is attached to OptoServer.
5. Start Visual Basic and open the OptoSrvr\Examples\Vb5\Dde_tst1.vbp example.
6. To run the program, choose Run > Start.

   When you start running the Dde_tst1.vbp program, start-up code found in the Form_Load event for the main form, DDE_TST1.FRM, is executed. A DDE warm link (a Visual Basic Notify) is established to OptoServer and a program timer is initialized. Notice that the form changes slightly and does not display the Poke and Request objects that were in the upper-right quadrant of the form. The Program Status displays the message "Stopped."



CreateTables ()

ClearValues()

7. From the Poking menu, choose Start.

   The project begins and the Program Status changes to "Waiting." Values are poked out to the controller and then read back. The Item Names and Values are displayed along with changes to their respective bar graphs. You can change the Poke Interval to increase or decrease the time between pokes and requests.
8. To stop the example, choose Poking > Stop.

You've now run the sample Dde_tst1.vpb program that makes basic DDE pokes and requests from a controller and updates a Visual Basic form.

Viewing the Main Components of the Visual Basic Sample Program

To take a closer look at the objects associated with the Visual Basic program, we'll start by viewing the contents of the form. A form is a collection of objects, each with associated Visual Basic code that can be viewed by double-clicking the object. The View Form button displays the .frm file, and the View Code button displays Visual Basic code for project components.

1. After running the sample program, go back to the design programming environment by choosing Run > End.

   The DDE Test Program form is displayed for editing or viewing.

2. Double-click the timer graphic in the upper-right section of the form.

   The code window for this object appears. Most of the Visual Basic code for this example is found in this object's code module (DdePokeTimer).

3. View the code by scrolling through it. Click the Object drop-down list to see other code modules and objects. Close the code window.
4. Double-click other objects displayed in the form.

   Each object's code window is displayed. Notice that the Properties window shows information about the selected object. Close each Code window with its control-menu box.

5. Click the Request text box.

   This object senses changes to the OptoServer data (warm link) and requests the new data from OptoServer about the controller. Data is returned in a comma-delimited form. The code parses the information and displays it in the text boxes on the form. It also changes the message displayed in the Program Status text box.

OptoServer and OPC

Introduction

This chapter discusses how to use OptoServer to communicate with other OPC-compliant applications. OLE for Process Control (OPC) is a standard specifically designed for control applications. It uses OLE (Object Linking and Embedding) technology for fast data exchange and increased compatibility among servers and clients from different vendors. OptoServer supports OPC Specification 1.0 clients.

In This Chapter

General OPC Concepts

What Is OPC?

OPC is a technology that defines standard objects, methods, and properties for communication among field devices, their servers, and the software applications that use their information. These applications can include both automation and control software, such as OptoServer, and business and office software.

Traditionally, software vendors have provided specialized drivers to communicate with the varied hardware on the market. Since there has been no common standard, each software driver is specific to a piece of hardware and doesn't work for any other piece. When the hardware changes, the software vendor must choose whether to modify the driver to match. Drivers typically address basic functions and don't necessarily utilize all the features of the hardware.

Hardware vendors traditionally don't produce drivers, even though they know about the special features of their hardware and how best to access the information that hardware produces. If they did produce drivers, each piece of hardware would have to have multiple drivers, one for each specific software application used with it.

OPC is designed to do two things:

- First, create a common standard so all communication between software and hardware is in the same language.
- Second, move driver creation to the hardware manufacturer, who best understands the hardware's capabilities.

OPC technology has several advantages over DDE: OPC is quick, robust, designed for networks, and has a standard message syntax for all clients.

As the OPC standard gains wide support among developers, you will be able to choose among a wide selection of products, applications, and tools from different vendors, with the confidence that they will all work together seamlessly.

Example of OPC in Action

Without OPC: You want to use another vendor's human-machine interface (HMI) with an Opto 22 controller. You cannot use the HMI software unless it has a driver written specifically for Opto 22.

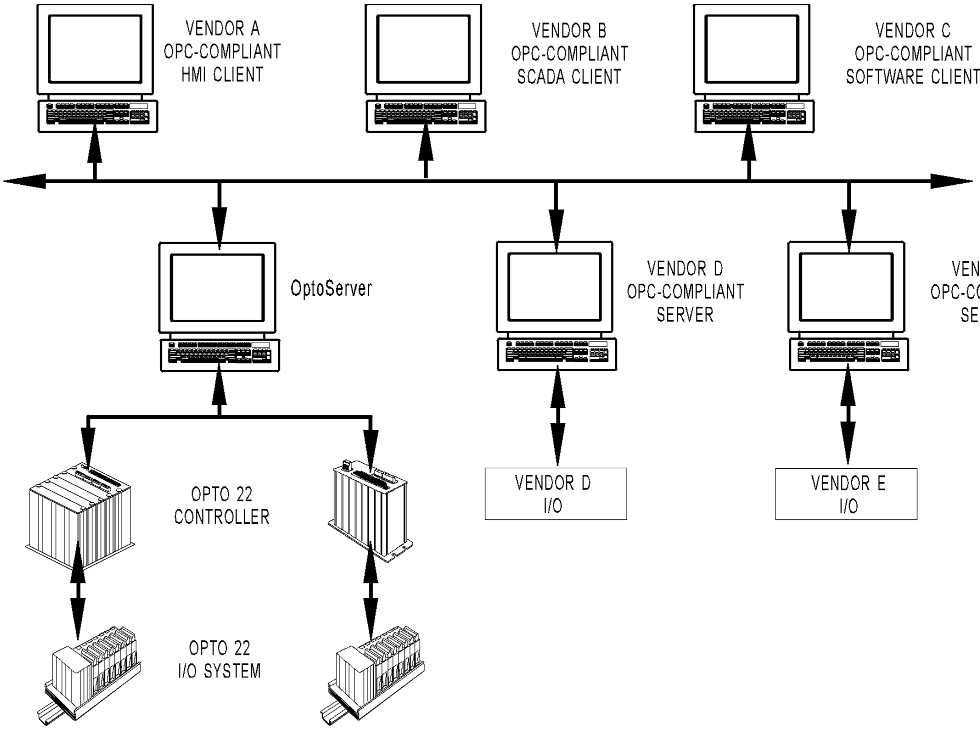
With OPC: If the other vendor's software is OPC 1.0-compliant, you don't need a separate driver. For example, suppose you have an Opto 22 controller running an OptoControl strategy. Since OptoServer is OPC-compliant, it can retrieve that data and provide it to the OPC client.

The OPC Specification

The OPC Foundation, a non-profit group, released the first OPC specification in August 1996, and OptoServer supports clients compliant with the 1.0 specification. The initial specification's scope concentrates on basic data acquisition. It does not address historical data handling, alarming, or security. The specification provides the following interfaces:

- A fast, efficient custom COM interface for data exchange between plant-floor servers and OPC clients, using a compiled language such as C or C++.
- A simplified and somewhat slower OLE Automation interface (on top of the COM interface) for accessing data within any OPC-compliant server using programs in Visual Basic, Delphi, and other languages that support OLE.

OPC supports a distributed architecture, allowing clients and servers to reside on different machines, transparently. The following figure illustrates a network using OPC and OptoServer.



Creating Client Links

The OPC client communicates with the server through standard links you create. Unlike DDE, OPC syntax rules do not vary by client.

The client link consists of three parts, or expressions: Server, Access Path, and Item ID. These expressions are basically equivalent to the DDE message expressions Application, Topic, and Item, explained in Chapter 3.

Server

The first part of the client link is simply the name and location of the server.

- **Name**—The OPC program ID is Opto22.Mds.1. It will probably appear in a drop-down menu or list box.
- **Location**—If the server you want to talk to is local (on the same machine), leave the Node Name field blank. If the server you want to talk to is remote (on another machine), enter the name of that machine in the Node Name field.

Access Path

The second part of the client link dictates the route that a server should take to get to the data requested. For OptoServer, you should normally use OPTO_MDS_0.

The access path is not required by OptoServer, however. If you are using a third-party software package that does not have a way to set it, it can be left out.

OptoServer will also create topics based upon the controller information you have set up. These topics will be in the following format:

OptoCtrl.ControllerName

where ControllerName is the name you supplied for the controller when it was configured.

Item ID

The third part of the client link is a string that pertains to the specific piece of data you're requesting. The string normally includes the controller name, scan rate, item type, item name, and additional fields, depending on the item you're accessing.

NOTE: The item name (or tag name) must be in all uppercase, and you must have already defined it in the OptoControl strategy. OptoControl requires all uppercase letters.

Item ID String Format

```
ControllerName[RefreshGroupName]:ItemType.ITEM_NAME[start-end].Field.BITn
```

Item ID String Samples

```
Mistic1[Group 0]:F.GEN_FLOAT_00
Mistic1[Group 0]:I.GEN_INT_00
[Group 0]:DIM.D_IN1.State
```

NOTE: If the Item ID string does not contain the controller name, OptoServer will look for it in the Access Path.

For details about the format of the Item ID string, see the Item Parameters table on page 31 and the Item Types table starting on page 32. Parameter descriptions and items supported are the same for DDE and OPC.

Also read "Setting Refresh Times" on page 22 and "Optimizing OptoServer Scanning" on page 19. These sections apply to OPC-compliant applications as well as to other OptoServer clients.

Running a Sample OPC Client

This section describes the basics of setting up OptoServer to provide information to an OPC client. In a typical setup, OptoServer and the OPC client would probably be running on different computers attached to the same network; but for this example we'll run them on the same computer. These are the steps we will follow in this section:

- Use OptoControl to download a strategy to the controller and verify communication.
- Use the OptoServer Administrator to configure a controller attached to the computer running OptoServer.
- Start OptoServer so that it is ready to be accessed by the OPC client.
- Start the OPC client and create client links.

Downloading a Strategy to the Controller and Verifying Communication

1. Start OptoControl.
2. From the File menu, choose Open Strategy. Navigate to Opto 22\FactoryFloor\OptoSrvr\Examples\OptoCtrl and open the sample strategy dde_test.cdb. (If OptoServer was not installed in the default directory, your directory will be different.)

   This strategy can be used to demonstrate an OPC client as well as a DDE client.
3. Configure the strategy for your controller's connection, address, and port settings, and enter "mistic1" as the controller's name.

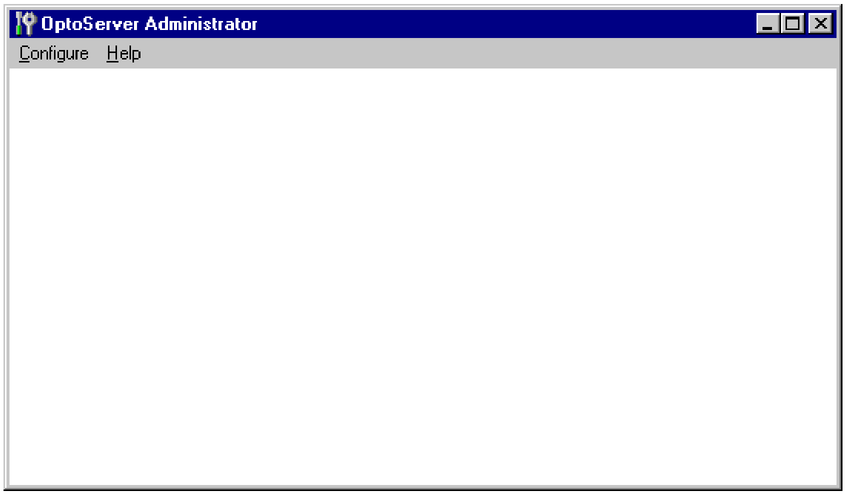
   If you need help, see "Configuring Controllers" in the *OptoControl User's Guide*.
4. Compile the program.
5. Change to the Debug mode and run the program.
6. Verify that the controller is communicating with your PC. Observe whether the controller's host port LED (ARCNET, COM0, etc.) flashes in response to messages from the PC.

   If your controller is responding to your PC with no errors, you're ready to move on to the next step. If you're having problems getting your controller to respond to the OptoControl strategy, make sure the configuration setup and cable wiring between the controller and PC are correct. Consult the *OptoControl User's Guide* and your controller manual to help you troubleshoot the problem.
7. When the controller is communicating with the PC, leave the strategy running and exit OptoControl.

Using OptoServer Administrator to Configure a Controller

1. Start OptoServer Administrator by clicking the Start button and selecting Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer Administrator.

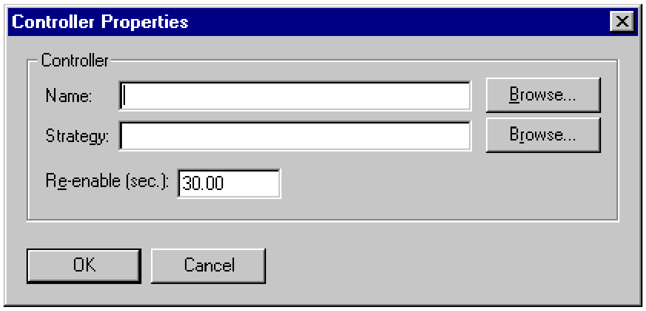
   The OptoServer Administrator main window appears:



2. Select Configure > Controllers.

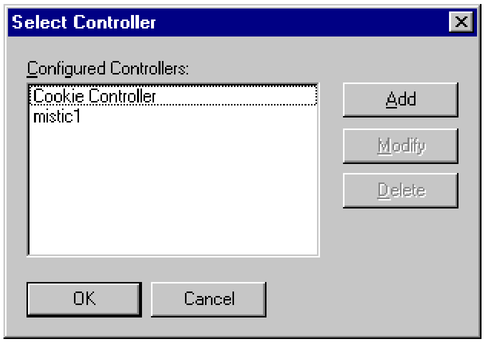
   If this is the first time a controller is being configured in OptoServer Administrator, a message about an INI file pops up. Click OK to continue. The necessary file is created.

3. In the Controllers dialog box, if mistic1 is already listed as a controller, skip to step 8. If it is not listed, click the Add button to add a controller.



Re-enable (sec.): 30.00

4. Click the Browse button to choose a controller from the Select Controller dialog box.



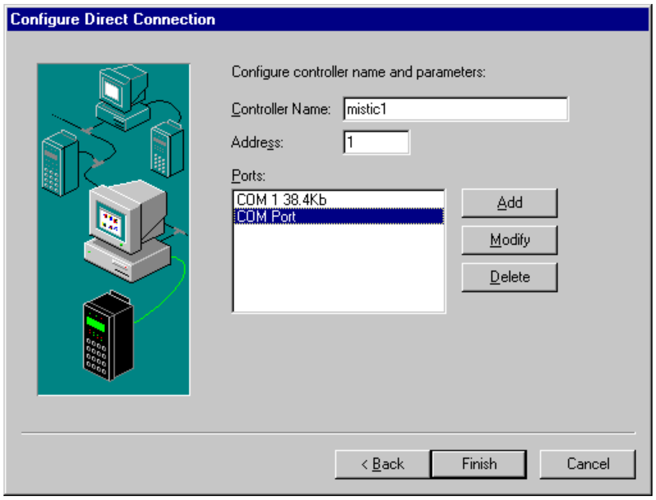
Configured Controllers:
Cookie Controller
mistic1

5. Click the mistic1 controller name you set up in the OptoControl strategy. Click Modify to check the controller's configuration.

   The type of connection should be shown as Direct.

6. Click Next.

   The Configure Direct Connection dialog box appears, with the mistic1 controller listed in the Controller Name box.


Configure Direct Connection

Configure controller name and parameters:

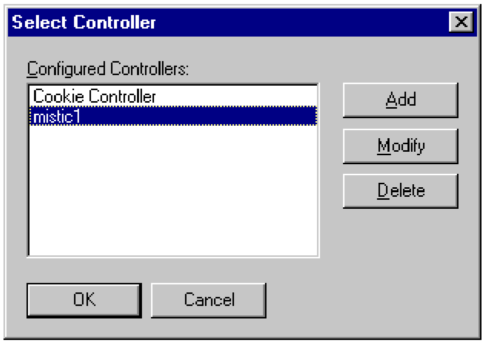
Controller Name: mistic1

Address: 1

Ports:
COM 1 38.4Kb
COM Port

7. If necessary, change the controller's address in the Address box. Highlight the port for direct connection to your controller. If you need to add a port, click Add. (For help, see "Configuring Controllers" in the *OptoControl User's Guide*.) When the address and port are correct, click Finish.

   The newly configured controller appears in the Select Controller dialog box.


Select Controller

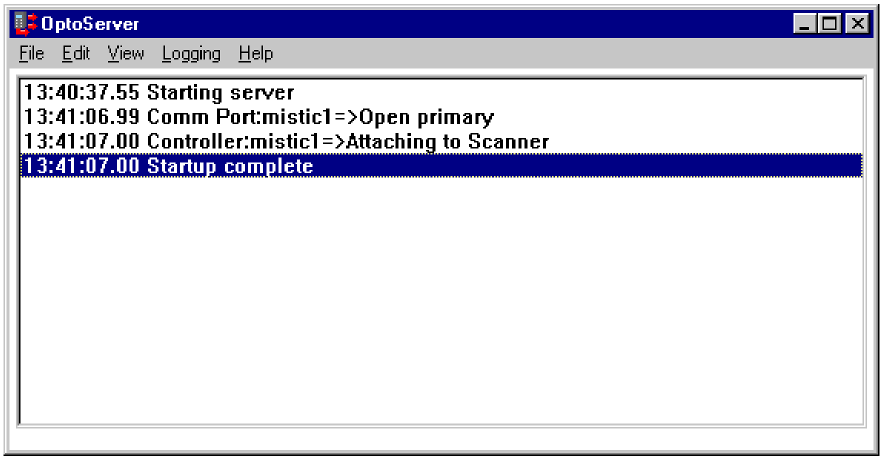
Configured Controllers:
Cookie Controller
mistic1

8. Click the mistic1 controller to highlight it, and click OK.
9. Click OK to close any open dialog boxes.
10. Set scan rates if you wish by following the instructions in the section "Setting Refresh Times" on page 22. Then return to the next step.
11. Exit OptoServer Administrator by clicking the close box.

Starting OptoServer

1. Click the Start button and select Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer.

   The OptoServer main window appears:



```
13:40:37.55 Starting server
13:41:06.99 Comm Port:mistic1=>Open primary
13:41:07.00 Controller:mistic1=>Attaching to Scanner
13:41:07.00 Startup complete
```

Your screen may show OptoServer attaching to other controllers in addition to mistic1, if you have configured others in OptoServer Administrator.

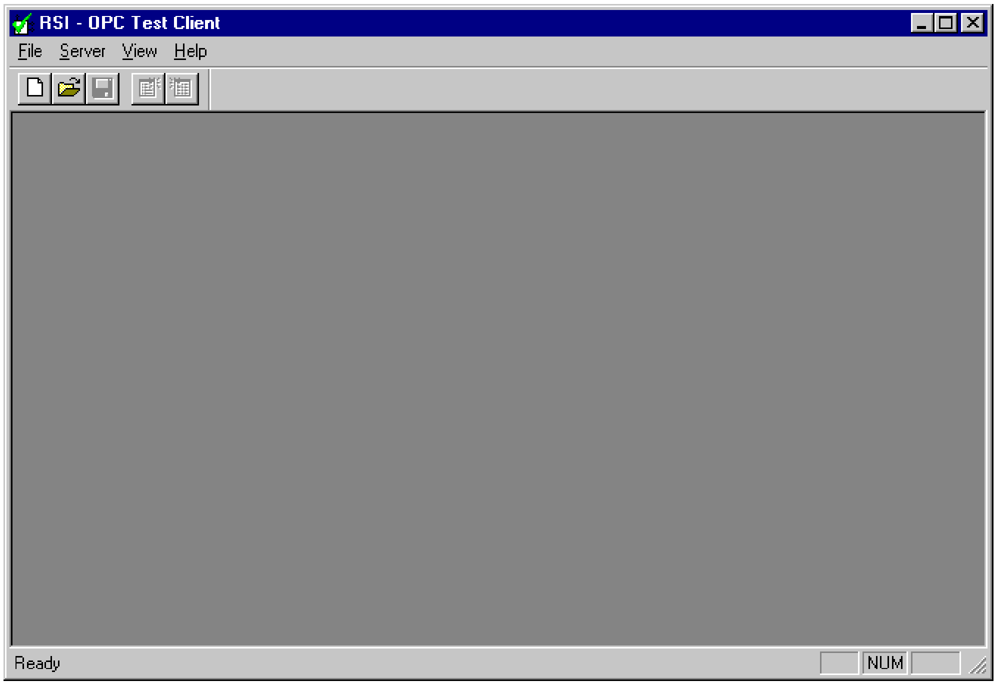
2. Click the Logging menu and make sure the following logging options are checked:
   - Log to Window—The main window displays messages.
   - Log Level Normal—Only significant OptoServer communication events are recorded.
   - Log with Time—Messages are saved with a time stamp.
3. Click the minimize button to reduce the OptoServer window to an icon.

Starting the OPC Client

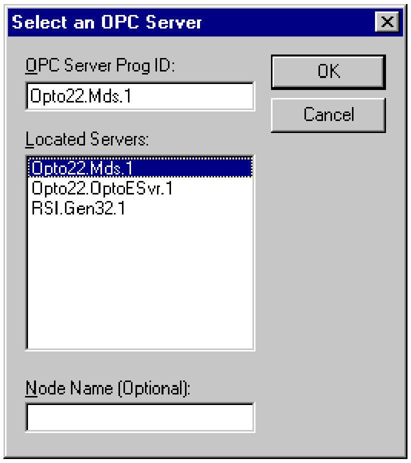
1. From the Start menu, choose Programs > Opto 22 > FactoryFloor 4.0 > OptoServer > OptoServer OPC Test Diagnostic.

   If you receive an error message about a file that’s missing, can’t be found, or is invalid, see “If You Are Running Windows 95” on page 50.

   The client’s main window appears:



2. From the Server menu, choose Connect.



OPC Server Prog ID: Opto22.Mds.1

Located Servers:
Opto22.Mds.1
Opto22.OptoESvr.1
RSI.Gen32.1

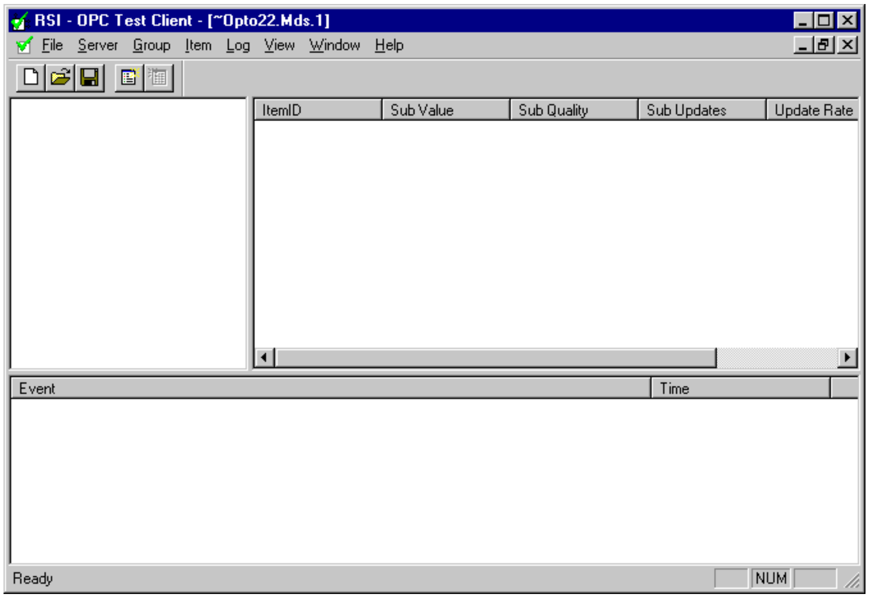
Node Name (Optional):

If OptoServer is already running, Opto22.Mds.1 appears in the list. If OptoServer is not already running, the client automatically starts it. If the server does not appear in the list, you can type its name in the OPC Server Prog ID field.

NOTE: In this example, OptoServer is running on the same computer as the OPC client, so you would leave the Node Name field blank. To run OptoServer on a different computer, you must first configure DCOM to recognize the other computer. See "If You Are Running OptoServer and an OPC Client on Different Computers" on page 50. Then in the previous figure, you would enter the name of the computer running OptoServer in the Node Name field.

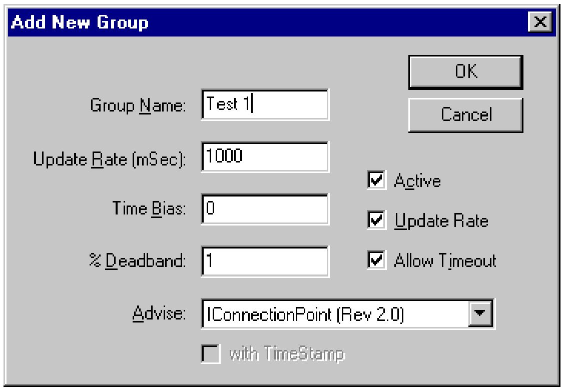
3. Highlight Opto22.Mds.1 and click OK to connect to OptoServer.

You return to the client's main window.


RSI - OPC Test Client - [~Opto22.Mds.1]
File Server Group Item Log View Window Help
ItemID
Sub Value
Sub Quality
Sub Updates
Update Rate
Event
Time
Ready
NUM

4. From the Group menu, choose Add Group.

   The client requires that you designate a "group" for the items the client is requesting (usually I/O points or variables). This group has no relationship to groups you may have set up in the Refresh Times dialog box.



Group Name: Test 1

Update Rate (mSec): 1000

Time Bias: 0

% Deadband: 1

☑ Active

☑ Update Rate

☑ Allow Timeout

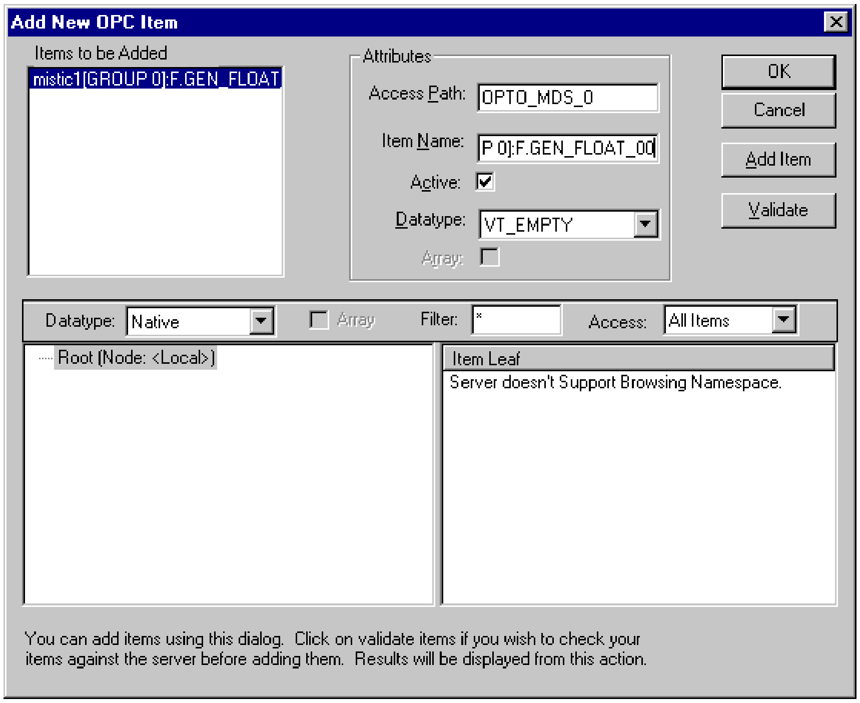
Advise: IConnectionPoint (Rev 2.0)

☐ with TimeStamp

5. Type in a Group Name (if the client requires it) and click Update Rate to put a check in the box, as shown in the previous figure. Click OK.

   The new group appears in the main window.

6. From the Item menu, choose Add Item.



Add New OPC Item

Items to be Added

mistic1[GROUP 0]:F.GEN_FLOAT

Attributes

Access Path: OPTO_MDS_0

Item Name: P 0]:F.GEN_FLOAT_00

Active: ☑

Datatype: VT_EMPTY

Array: □

Datatype: Native   □ Array   Filter: *   Access: All Items

Root (Node: <Local>)

Item Leaf

Server doesn't Support Browsing Namespace.

You can add items using this dialog. Click on validate items if you wish to check your items against the server before adding them. Results will be displayed from this action.

7. In the Access Path field, type OPTO_MDS_0 as shown in the figure. In the Item Name field, type the following:

```
mistic1[GROUP 0]:F.GEN_FLOAT_00
```

What you just typed reflects the format for the Item ID, described in detail on page 42:

```
ControllerName[RefreshGroupName]:ItemType.ITEM_NAME[start-end].Field.BITn
```

Note, however, that the client inserts the apostrophes for you, so you don't have to type them. In this example, we've used the default name for the RefreshGroupName. If you entered a different name for the refresh group, use the name you entered instead of GROUP 0.

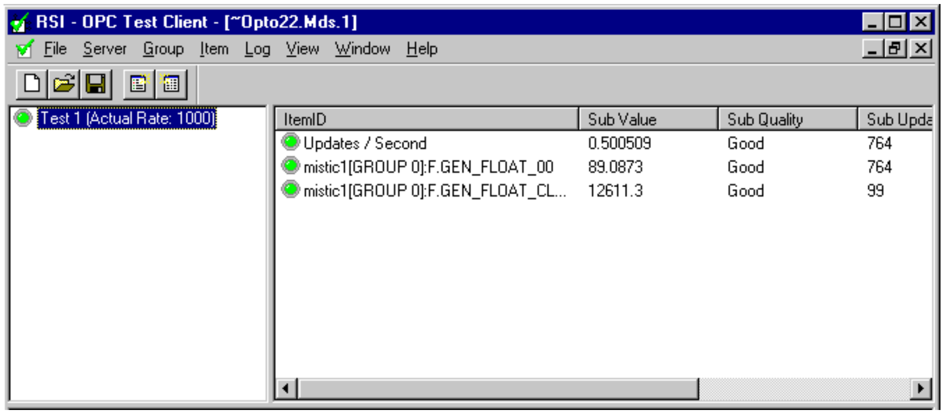
For the GEN_FLOAT_00 variable in this example, [start-end], Field, and BITn are not applicable.

8. When you have entered the Access Path and the Item Name, click the Add Item button.

The item is added and the Item Name field is cleared. You can add another item the same way if you wish.

9. When you have finished adding items, click OK.

The main window shows the items you have added. The following figure shows two items, GEN_FLOAT_00 and GEN_FLOAT_CLOCK, which are both variables from the strategy:



● Test 1 (Actual Rate: 1000)

ItemID	Sub Value	Sub Quality	Sub Upda
● Updates / Second	0.500509	Good	764
● mistic1[GROUP 0]:F.GEN_FLOAT_00	89.0873	Good	764
● mistic1[GROUP 0]:F.GEN_FLOAT_CL...	12611.3	Good	99

The sample connection is complete! You can watch the items being updated at the refresh rate you specified.

If You Are Running Windows 95

If you are running OptoServer on Windows 95, read this section.

You must have installed distributed COM (DCOM) in order to run OPC. DCOM comes standard with Windows NT 4.0 and Windows 98 and is available as an add-on for Windows 95. DCOM also comes with Internet Explorer 4.0. The DCOM package can be obtained from Microsoft's Web site at www.microsoft.com/com/dcom95.

If you are running OptoServer and an OPC client on different computers that are networked, rather than on the same computer, install DCOM on both computers. Additionally, you must install the DCOMcnfg program. It is also available from the Microsoft Web site.

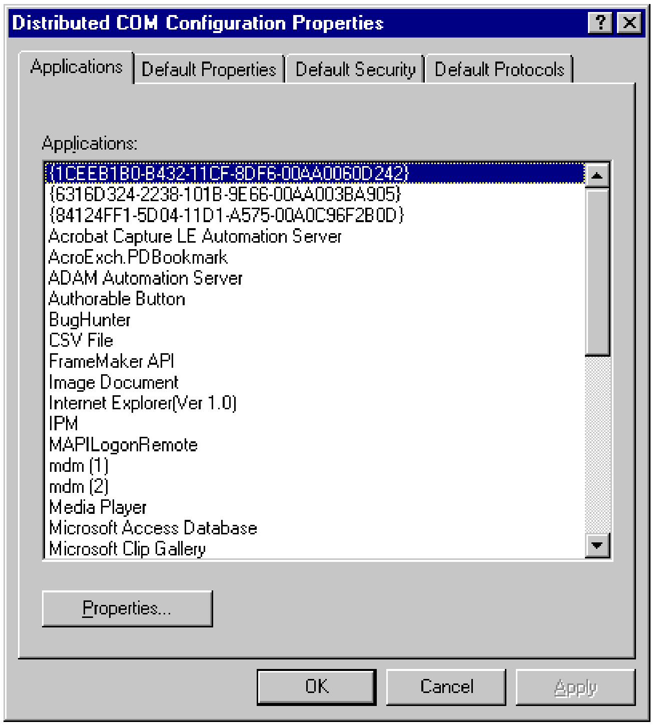
Also note that if you are running OptoServer on a Windows 95 computer and the OPC client on a separate computer, you must manually start OptoServer. The client does not start it remotely.

If You Are Running OptoServer and an OPC Client on Different Computers

If you are running OptoServer and an OPC client on two separate networked computers, you'll need to configure DCOM to recognize the computer running OptoServer.

1. On the client computer, open Windows NT Explorer (Windows Explorer on Windows 95 or Windows 98) and navigate to the WINNT\system32 directory (Windows\System directory on Windows 95 or Windows 98). Locate DCOMcnfg.exe and double-click it to execute the program.

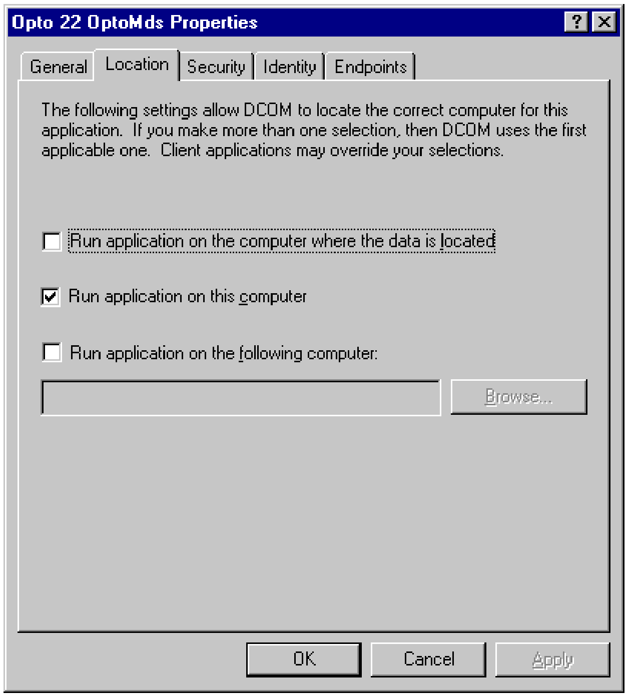
   The Distributed COM Configuration Properties window appears:



Applications:

{1CEEB1B0-B432-11CF-8DF6-00AA0060D242}
{6316D324-2238-101B-9E66-00AA003BA905}
{84124FF1-5D04-11D1-A575-00A0C96F2B0D}
Acrobat Capture LE Automation Server
AcroExch.PDBookmark
ADAM Automation Server
Authorable Button
BugHunter
CSV File
FrameMaker API
Image Document
Internet Explorer(Ver 1.0)
IPM
MAPILogonRemote
mdm (1)
mdm (2)
Media Player
Microsoft Access Database
Microsoft Clip Gallery

2. Scroll down and double-click Opto 22 OptoMds.
3. When the Properties window opens, click the Location tab.



The following settings allow DCOM to locate the correct computer for this application. If you make more than one selection, then DCOM uses the first applicable one. Client applications may override your selections.

☐ Run application on the computer where the data is located

☑ Run application on this computer

☐ Run application on the following computer:

4. Check the box next to *Run application on the following computer*. Type in the name of the computer that will be running OptoServer, or click the Browse button (if available) to locate the computer. When the computer's name appears in the field, click OK.

5. Click OK in the window to exit DCOMcnfg.

DCOM is now configured to recognize the computer running OptoServer. When you start the client and connect to the server, make sure to type the computer name in the Node Name field. See the figure on page 47 for an example of a client dialog box requesting Node Name. If you leave the Node Name field blank, the server defaults to the local computer.

Also note that if you are running Windows 95 on the OptoServer computer, you must start OptoServer manually. The client does not automatically start a remote server.

OptoServer Troubleshooting

Introduction

If you have problems while running OptoServer, check the troubleshooting tips in this appendix and also the messages starting on page 56. For help on using OPC clients with OptoServer, see "Problems Using OPC" on page 55. To display additional diagnostic messages that may help troubleshoot your system, choose Logging > Log Level Diagnostics in OptoServer. For additional help, call Opto 22 Product Support. See page ix for a list of service phone numbers.

Communication Problems

If you are a Windows NT user and are experiencing communication problems that are indicated by the errors listed below, first try rebooting your machine to ensure that all port configurations have been correctly registered with the operating system.

- "Invalid port error. WinRT drivers might not have started."
- "Invalid protocol error."
- "Port setup invalid."
- "Could not find other nodes in ARCNET."

If you are using ARCNET, make sure there are no Windows 95 or Windows NT network drivers configured to use the ARCNET card. These drivers conflict with Opto 22 ARCNET drivers.

You can use OptoTerm to test communication to controllers. When OptoTerm starts, it displays a list of controllers configured on your computer. Double-click a controller name to display a status dialog box. For more information about OptoTerm, see the Troubleshooting appendix in the *OptoControl User's Guide*.

Network Communication

To troubleshoot DDE communications between OptoServer and a DDE-aware client, try swapping out the server or the client with another DDE-aware client such as Word or Access. Example: OptoServer and a Visual Basic client are not "talking" to each other. If swapping out Visual Basic with Access shows communication is fine between Access and OptoServer, there may be a problem with the Visual Basic application.

View the OptoServer main log window to see OptoServer transaction and error messages. Select Logging > Log Level Diagnostics to display all messages sent to and received from clients. Messages displayed under Log Level Normal conditions are explained starting on page 56.

To see if a client is attached to OptoServer, choose the OptoServer command View > Clients. The Client List dialog shows an eight-digit client handle number for each connected DDE-aware client, and the node name for each connected OptoDisplay client.

To view communication exchanges at the controller level, run the OptoSniff utility that's included with FactoryFloor. From the main window, choose the File > Enable Sniff command. Anything sent or received by the controller is displayed in the main window. You can also use the utility as a terminal emulator to send and receive from the controller. For more information on OptoSniff, see the Troubleshooting appendix in the *OptoControl User's Guide*.

To verify that a DDE-aware client works with OptoServer, download a simple OptoControl strategy to the controller and see if the client is being updated by OptoServer. An example of a simple OptoControl program is a one-chart, one-variable program that increments the variable in an infinite loop.

Make sure you do not install a Windows network driver on the ARCNET card in your PC at the time you install a Windows programming environment. Operating systems such as Windows 95 and Windows NT automatically try to install a network driver on an ARCNET card. Allowing the driver to be installed can produce unpredictable results from OptoServer.

Verify that the "Retries" port parameter is not masking other problems. Set the retries parameter to zero and find out exactly what happens by analyzing whether OptoServer is not responding, whether there are long delays, or whether it's getting a sequence I/O error.

Verify that there are no Windows Control Panel COM port conflicts. Make sure there are no conflicts between the ports configured in the Windows Control Panel and in OptoServer. Check that the address IRQ settings are unique for each port and that they match the settings used in OptoServer Administrator. If communication problems with OptoServer remain, try using OptoTerm. (See the Troubleshooting appendix in the *OptoControl User's Guide* for information on OptoTerm.)

Lengthen timeout per port. This action should help eliminate some of the common error messages.

Understanding Timeout, Retries, and Re-enable Settings

OptoServer polls controllers sequentially, which means that it normally waits for a response before sending out another command. If the controller does not respond within the timeout period, the retries number determines how many times OptoServer will try again. If all the tries fail, you receive a timeout error, and OptoServer checks the re-enable setting to see how soon to try communicating with the controller again.

For example, if the timeout period is set to 3 seconds, retries are set to 1, and the re-enable setting is 30 seconds, here's what will happen:

> *OptoServer polls the controller and waits for a response. If it does not receive a response within 3 seconds (timeout period), it polls the controller again (one retry). If it still doesn't receive a response, you see a timeout error. OptoServer waits for 30 seconds (the re-enable setting), and then starts the sequence over again.*

During the timeout/retry cycle, OptoServer does not communicate with any other controllers. The computer may become sluggish or unresponsive, making it difficult to access menu options and other applications. During the re-enable period, the system returns to normal: OptoServer polls other controllers, and the computer functions normally. Therefore, if the timeout period is long, if the retries number is large, or if the re-enable period is too short, system performance may suffer during communication problems.

Timeouts, retry numbers, and re-enable periods are set when you configure the controller. For timeouts, the best values to use depend on your method of communication. Timeout values that are too short result in unnecessary retries; timeout values that are too long cause unnecessary delays when there are communication problems. The following table shows basic guidelines:

Communication Method	Timeout Value
ARCNET	1–2 seconds
Ethernet	3–5 seconds
Serial at 115.2 Kbaud	2–3 seconds

The default of one retry is reasonable unless the communications link is susceptible to errors or noise, as with some modems.

The re-enable period should generally be set no less than 30 seconds (the default). If OptoServer is polling multiple controllers, set the re-enable period for each controller long enough to allow normal polling of the other controllers in case one goes offline.

Problems Using OPC

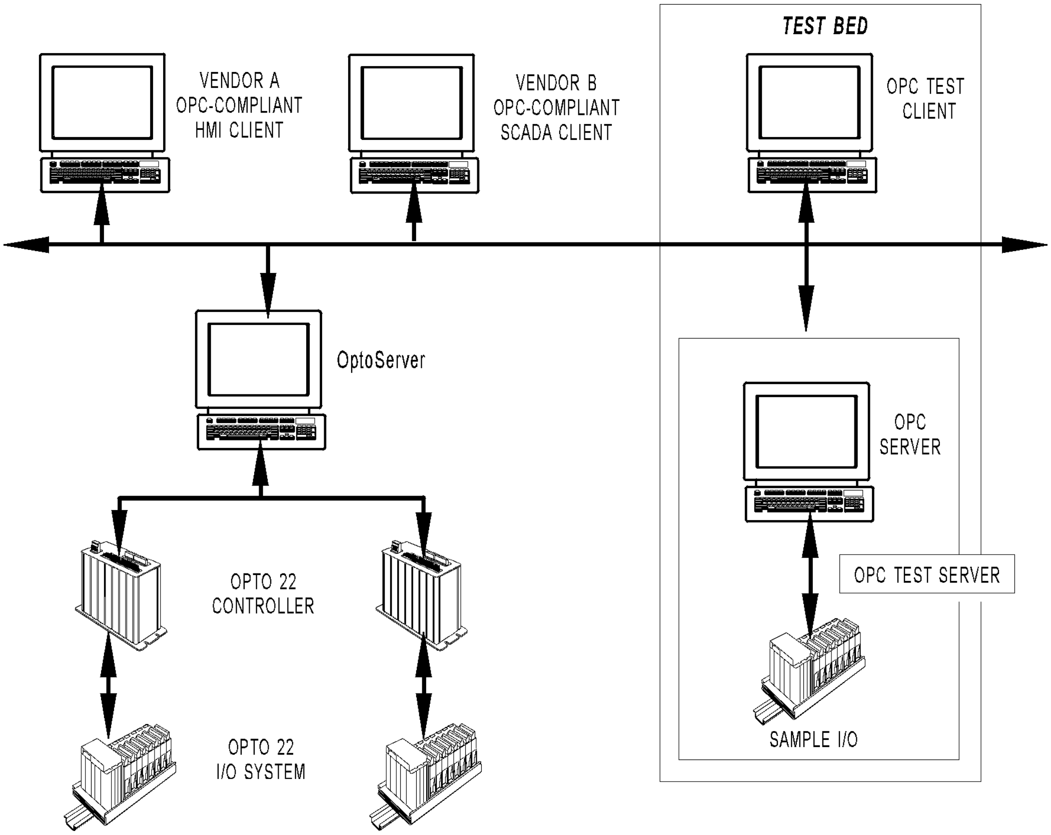
Necessary Files

You must have DCOM installed to run OptoServer. If you are running Windows 95 and have OptoServer and the OPC client on different computers, you must also install the DCOM configuration utility, DCOMcnfg.exe. See "If You Are Running Windows 95" on page 50 and "If You Are Running OptoServer and an OPC Client on Different Computers" on page 50 for more information on DCOM and DCOMcnfg.exe.

Using the Diagnostic Utilities

Included in OptoServer are two test applications: OPC Generic, a server, and OPCTest, a client. If you are having trouble connecting your client to OptoServer, you can test your client with the Generic server, or you can test the OPCTest client with OptoServer. Both the server and the client are located in the Diagnostic directory.

The following figure shows how the test bed works in relation to OptoServer.



Messages

This section describes messages that may appear in the OptoServer (Mds) log window. The Communication Data Server does not display messages except for "Initializing" and "Ready" messages that appear in the status bar at the bottom of its window.

Because these messages are from Mds, any client they refer to is an OPC or DDE client. <TagName> refers to an item or item ID expression such as MyRtu[Group 0]:I.MY_INTEGER, where MY_INTEGER is the name of a variable in a strategy on the controller MyRtu. Several messages include a <Reason>, which describes the problem in more detail to help you solve it. If you need additional help in solving errors, contact Opto 22 Product Support. (See page ix for contact information.)

Both error messages and typical transaction messages are included in this section. Messages are listed in alphabetical order.

A client has connected. <Qty> client(s are) connected. Status message indicating that a client has just connected to OptoServer and how many clients are now connected.

A client has disconnected. <Qty> client(s are) connected. Status message indicating that a client has just disconnected from OptoServer and how many clients are now connected.

Adding item/tag <TagName>. Indicates that the item <TagName> is being referenced by a client.

Advise loop start failed: <Reason> <TagName>. A hot link or report-by-exception subscription failed. Possible reasons for failure include an incorrect controller name or an invalid tag name. See <Reason> for a more detailed explanation.

Could not start <Exe>. One of the three executable files that OptoServer (Mds) automatically starts could not be started. If you don't need the file, you can ignore the message, since Mds will continue to run. If you need the file, make sure it is present. These are the three files:

- NetDDE, used for DDE links over a network. Mds starts NetDDE unless it's already running.
- OptoCds.exe, the portion of OptoServer used by OptoControl, OptoDisplay, and any other client using OptoCom.dll.
- OptoSh16.exe or OptoSh32.exe, needed to create the DDE "shares" that allow NetDDE connections to be made.

Destroy item/tag <TagName>. Indicates that the item <TagName> is no longer being referenced by any client.

Destroy topic <TopicName>. Indicates that the topic <TopicName> is no longer being referenced by any client. (An example of a topic name would be OPTO_MDS_0.)

Error adding item/tag <TagName> <Reason>. A problem occurred while a client was referencing an item. See <Reason> for a more detailed explanation.

<Exe> already running. One of the three executable files that OptoServer (Mds) automatically starts has already been started. These three files are NetDDE, OptoCds.exe, and OptoSh16.exe or OptoSh32.exe.

Failure sending <TagName> to <Qty> client(s). Data failed to reach one or more clients. May indicate an internal error. Contact Opto 22 Product Support.

New item/tag <TagName>. Appears the first time the item <TagName> is referenced by a client.

New topic <TopicName>. A client attached to the server and referenced the topic <TopicName>. (An example of a topic name would be OPTO_MDS_0.)

Poke failed <Reason> <TagName>. A client attempted to perform a synchronous write. (DDE writes are called "pokes.") This message refers to a synchronous request, not an asynchronous or hot-link request, which is a report-by-exception data transfer. See <Reason> for an explanation of the failure.

Request failed <Reason> <TagName>. A client attempted to perform a synchronous read. This message refers to a synchronous request, not an asynchronous or hot-link request, which is a report-by-exception data transfer. See <Reason> for an explanation of the failure.

Request for item/tag <TagName>. A client successfully performed a synchronous read. This message refers to a synchronous request, not an asynchronous or hot-link request, which is a report-by-exception data transfer.

Sending <TagName> to <Qty> client(s). Data is being sent to one or more clients, either by client request or as a report-by-exception.

Started <Exe>. One of the three executable files that OptoServer (Mds) automatically starts has been started. These three files are:

- NetDDE, used for DDE links over a network. Mds starts NetDDE unless it is already running.
- OptoCds.exe, the portion of OptoServer used by OptoControl, OptoDisplay, and any other client using OptoCom.dll.
- OptoSh16.exe or OptoSh32.exe, which create the DDE "shares" that allow NetDDE connections to be made.

Start item/tag <TagName>. A client has referenced the item <TagName>.

Stop item/tag <TagName>. A client has stopped referencing the item <TagName>.

Write (poke) for item/tag <TagName>. A client successfully performed a synchronous write. (DDE writes are called "pokes.") This message refers to a synchronous request, not an asynchronous or hot-link request, which is a report-by-exception data transfer.

Problems with Windows Permissions

When you set up controllers on a computer running the Microsoft Windows 2000 or Windows XP operating systems, typically you are using the computer with top-level "administrator" privileges. If someone later uses this same computer to run a FactoryFloor application, but logs in to the computer with lower-level, non-administrator privileges, the FactoryFloor application may not recognize controllers that have been previously configured.

If this problem occurs, you can modify the Windows permissions to let specific users access previously configured controllers without having administrator access. This is done using the Registry Editor utility.

WARNING: *Use the Windows Registry Editor carefully. It is strongly recommended that you make a backup copy of your Windows Registry before continuing with this procedure. Without a backup copy, if you delete the wrong properties and cannot return the Registry to its original state, application and system files can become unusable and will have to be reinstalled.*

1. From the Windows Start menu, select Run.

   The Run dialog box appears.

2. Enter the following command in the Open field and press ENTER:

   ```
   regedt32
   ```

   The Registry Editor main window appears with several open windows inside it.

3. Select the HKEY_LOCAL_MACHINE window to make it active.
4. Double-click the Software folder in the HKEY_LOCAL_MACHINE window.
5. Select the Opto22 folder.
6. Select Security?Permissions.

The Registry Key Permissions dialog box opens. Make sure that “Opto22” appears next to Registry Key at the top of the window.

7. Click Add.
8. In the Add Users and Groups dialog box, select the name of the appropriate group or domain from the List Names From drop-down list.
9. In the Names list, select the name of the user or group that will get controller access and then click Add.
10. If it is not already selected, choose “Full Control” from the Type of Access drop-down menu.
11. Click OK.
12. In the Registry Key Permissions dialog box, select the Replace Permission on Existing Subkeys checkbox and click OK.
13. Select Registry?Exit to close the Registry Editor.
14. Restart the computer.

The user or group you added can now use controllers without having administrator access.

Problems with Windows XP, Service Pack 2

When you configure a controller to use the *server* connection type on a computer running Microsoft Windows XP with Service Pack 2, you might not be able to configure the controller. If this occurs, you can work around the problem by modifying settings in the Group Policy console.

1. From the Windows Start menu, select Run.

   The Run dialog box appears.

1.In the Open field, type `gpedit.msc` and then press ENTER:

   The Group Policy console opens.

2. Using the console tree in the Group Policy manager, navigate to Computer Configuration > Administrative Templates > System > Remote Procedure Call.
3. In the Setting column, double-click RPC Troubleshooting State Information to display the properties dialog for this setting.
4. Select Enabled, and then select None from the drop-down list.
5. Click OK to close the dialog.
6. In the Setting column, double-click Restrictions for Unauthenticated RPC clients to display the properties dialog for this setting.
7. Select Enabled, and then select None from the drop-down list.
8. Click OK to close the dialog.
9. Shut down all programs and then re-boot your PC.

After re-booting, you should be able to access your controllers through an OPTOCDS connection.

OptoServer Files

This appendix provides a list of OptoServer files. Use this information as a reference when you are looking through your OptoServer directory.

IC32CKIT.dll	File essential for OPC and DDE connectivity
Mdslog.log	Default name of the message logging file created by OptoServer (Mds.exe)
OptoCds.exe	Communication Data Server executable program file used to communicate to OptoControl, OptoDisplay, or any other application using OptoCom.dll, such as Microsoft Visual Basic applications you may have written
OptoCom.dll	File in the system directory necessary for communication with controllers
OptoErr.dll	File in the system directory that contains a list of error numbers and messages
OptoMds.ini	Initialization file created by OptoServer Administrator when it is exited for the first time
OptoMwd.dll	File in the system directory necessary for communication with ports
Mds.exe	OptoServer executable program file used to communicate to OPC-compliant and DDE-aware clients
Mdshelp.hlp	Help file for OptoServer
OptoSa.exe	OptoServer Administrator executable program file
Readme.txt	Latest readme file for this release

FactoryFloor Glossary

action blocks Rectangular chart blocks that contain commands (instructions) to be executed in an OptoControl™ strategy. Several commands can be placed in one block. Action blocks can have many entrances, but only one exit.

access path One of three elements (server, access path, and item ID) used by an OPC-compliant client in an OPC message to identify a particular piece of data from a server. Similar to a DDE *topic*.

adapter card A printed circuit board, often installed within a computer, used to transfer data between a bus and a device.

alarm graphic A graphic that displays the state of alarms in OptoDisplay™. It can be used to acknowledge alarms and to view alarm logs.

alarm point In OptoDisplay, an OptoControl tag with a defined alarm state. The point name can be the tag name or a user-selected name.

amp or ampere Unit of current, abbreviation A.

analog Describes data that can have a continuous range of values, such as current, voltage, or pressure readings.

analog point An input or output point that can have a range of values. Voltage and current modules are examples of modules with analog points.

API (Application program interface) A set of routines in a programming library used by a software application. For example, the Pamux driver provides APIs that allow easy access to the Pamux bus. Microsoft® Windows® also provides APIs for various purposes, such as accessing serial ports and displaying message boxes.

application In a DDE message, the name of the DDE server the information is being requested from. It is one of three elements (application, topic, and item) used by a client in a DDE message to identify a particular piece of data (such as an integer, a string, an I/O point, or a range of cells in a work sheet) from a server.

application manager A dynamic object used to launch another application based on changes in process values. Each application manager has a command line, a working directory, and an associated trigger. The program file specified in the command line is launched each time the trigger condition occurs.

ASCII (American Standard Code for Information Interchange) Developed by the American National Standards Institute (ANSI), ASCII is a set of 128 characters that include letters, numbers, punctuation, and control codes, each represented by a unique number.

ASCII mode See *communication mode.*

automation A means of adding intelligence to an industrial process. Automating a process decreases the need for active human participation in the process. It also improves the performance, accuracy, and reliability of a process.

B1 A brain board used to attach up to 16 digital I/O points to an Optomux serial network.

B100 A brain board used to attach up to 16 digital I/O points to a Mistic serial network; same footprint as a B1.

B2 A brain board used to attach up to 16 analog I/O points to an Optomux serial network. Uses Opto 22 Generation 1 I/O modules only.

B200 A brain board used to attach up to 16 analog I/O points to a Mistic serial network; same footprint as a B2. Uses Opto 22 Generation 1 I/O modules only.

B3000 A brain used to connect up to 32 digital and up to 32 analog SNAP channels to a Mistic or Optomux network. Also see *SNAP-B3000-ENET.*

B4 An addressable digital brain board that can control up to 32 input or output points in distributed I/O applications. Any combination of Pamux B4 brain boards may be linked on a single Pamux bus to control up to 512 points of analog and digital I/O.

B5 An addressable digital brain board that can control up to 16 input or output points in distributed I/O applications. Any combination of Pamux B5 brain boards may be linked on a single Pamux bus to control up to 512 points of analog and digital I/O.

B6 An addressable analog brain board that can control up to 16 input or output points in distributed I/O applications. Any combination of Pamux B6 brain boards may be linked on a single Pamux bus to control up to 512 points of analog and digital I/O. The B6 includes an on-board microprocessor that continually scans all I/O points on the mounting rack, performs necessary conversions, and then updates a dual-port RAM. The host computer transfers data along the Pamux bus by reading from or writing to the dual-port RAM.

bank In a Pamux system, a group of eight digital I/O channels. A 16-channel digital I/O mounting rack with a B5 brain board has two banks. If the brain board is at address 20 and the adapter card (for example, a PCI-AC28) is at base I/O address 100, then the two banks are at I/O addresses 120 and 121. Banks and points are used together to access I/O points on a board at a particular address.

base address The starting I/O address for programmable registers, used as the reference address for all other I/O addresses.

basic trend A rectangular dynamic object in OptoDisplay that graphs the change in a variable or set of variables over time. Trends show variables on the vertical axis and time on the horizontal axis. Up to four trend lines can be displayed on any one trend chart. The maximum time span supported by each trend is 14 days. Compare to *SuperTrend*.

baud rate The clock rate for serial data transmission. A 38.4 Kbaud device can transmit or receive information at a maximum of 38,400 bits per second.

binary Data transmitted or stored as a bit pattern, rather than ASCII characters. Binary data is more compact and loads faster than ASCII data.

binary mode See *communication mode.*

bit A single binary digit (0 or 1).

bitmap An electronic file of an image. Bitmap files have a file extension of .BMP. Bitmap files can be created using almost any paint program and can be imported into an OptoDisplay project.

Boolean Logical operations involving true and false, and the operators AND, OR, and NOT. Booleans are commonly used when constructing search queries.

brain A processor that plugs into a SNAP rack next to analog, digital, and/or special-purpose I/O modules, forming an intelligent I/O unit. In addition to communicating with a host computer or controller, brains provide the local intelligence necessary to perform basic control functions such as on/off control, counting, and latching, or complex tasks such as PID control, temperature conversion, time proportional output, and emergency shutdown. SNAP brains are available in a variety of communication links and protocols, including wired or wireless LAN Ethernet, serial (Modbus, Mistic, or Optomux protocols), Pamux, ARCNET (Mistic protocol), and Profibus-DP.

brain board A processor that connects an analog or digital I/O mounting rack to a communication bus, such as the Optomux or Pamux bus. Like brains, brain boards communicate with a host computer or controller and perform both basic control functions and complex tasks locally. Brain boards and brains differ in their form: brains are in an enclosure; brain boards are not.

bus Single common cable used to connect all devices on a system. The Pamux bus is a 50-pin flat-ribbon cable. Optomux and Mistic use RS-422/485 serial busses.

byte A group of eight bits; an eight-bit binary number. For example, 10011011. Eight bits of information composed of zeros or ones, one of which may include a parity bit. Most character sets, for example ASCII or EBCDIC, use one byte per character of information such as a letter, a number or digit, or sometimes a punctuation mark or a symbol, such as $. A byte is to a bit what a word is to a character. Sometimes a byte is called an octet.

cache In OptoServer™, a storage place for data scanned from controllers. A cache is used by the OptoServer Communication Data Server to reduce unnecessary or repetitive scanning. If requested data is already in the cache and is “fresh” enough, the server delivers it to the client immediately without having to contact the controller. Similar to a disk cache or Internet cache, it improves throughput because it’s much faster to retrieve data from the cache in memory than from the controller. Caching applies to OptoDisplay, OptoControl, and other applications that use the OptoCom.dll.

.cdb The file extension for an OptoControl strategy. CDB stands for controller database.

channel See *point*.

chart A series of instructions in the form of a flowchart in OptoControl. Sometimes called a task. Charts can include action blocks, condition blocks, OptoScript™ blocks, continue blocks, connections, and text blocks.

client A program running on a computer node that uses a service provided by a server. For example, file servers provide the service of a big hard disk. Clients use files provided by the server. DDE servers provide a source of data and can automatically inform clients when data changes. OptoServer clients include OptoDisplay and other Opto 22 applications, OPC-compliant applications, and any DDE-aware program.

color button A color button appears as a colored rectangle in dialog boxes. Its color indicates the color currently selected for the associated item in the dialog box. To change the color, select the Color button.

COM (Component object model) The core of *OLE* and *DCOM* technology; provides standard interfaces and inter-component communication, so an application can use features of another application object or operating system. Underlies much of the code developed for Windows 95 and Windows NT operating systems, either by Microsoft or by others.

communication mode The protocol used to represent the information packets exchanged between two devices, such as an Opto 22 controller and a PC. Two serial modes are available: ASCII and binary. In ASCII mode, command messages are transmitted as a series of ASCII characters. Use ASCII mode if there is a modem between the OptoServer PC and a connected controller, or if you want to view communication messages on an ASCII display terminal. In binary mode, messages are represented as binary data, which may use 9-bit bytes. Binary mode is typically used because it is a faster protocol.

compile In OptoControl, the process by which strategy instructions are interpreted into Forth language code that can be downloaded to a controller.

condition blocks Diamond-shaped chart blocks that contain questions that control the logical flow of an OptoControl strategy. Condition blocks can have many entrances, but only two exits: True and False. More than one condition may be evaluated within a single condition block. The operators AND and OR determine whether all conditions in a block must be true to exit true (AND) or whether only one of the conditions in a block must be true to exit true (OR).

Configurator (OptoDisplay) A program from the OptoDisplay package used to configure an OptoDisplay project. The Configurator is used to draw display windows and attach graphical elements in the window to *tags*, which are data elements in a controller’s OptoControl strategy.

connection Used to indicate a relationship between a graphic object in OptoDisplay and a tag name (I/O or variable) in an OptoControl strategy. The connection affects an object's dynamic attribute as tag data changes. The OptoControl program and OptoDisplay Configurator are tightly integrated to make the connection process easy by simply defining what controller strategies should be used.

In OptoControl, a line with an arrow that connects flowchart blocks and defines the logical path of a chart's execution.

constant See *literal*.

continue blocks Oval chart blocks that route the logical flow of an OptoControl strategy to an action or condition block. Continue blocks store only the name of the next block.

controller-driven dynamic attributes Connections made to a dynamic object that change the appearance of the object based on the value or state of a variable. For example, changes in tag values within a controller may cause graphics in OptoDisplay to change size, color, or shape.

conversation The exchange of *DDE* messages between a DDE *client* and *server*. A conversation is conducted through a channel. The syntax of the messages exchanged varies among DDE-aware programs, DDE link types, and the content of the data being transferred. DDE clients initiate a conversation; DDE servers respond to the DDE client request.

.csb The file extension for an OptoControl subroutine.

DCOM (Distributed component object model) Extends COM to networks; makes remote objects appear to be local. DCOM was built into Windows NT and Windows 98; it must be installed in Windows 95.

DDE (Dynamic data exchange) A standard way for Windows applications to exchange data at runtime. Using DDE, process or plant data can be put into general-purpose applications like Microsoft Excel. Programs can either be on the same PC or on different PCs in a network environment.

A DDE link is initiated by a client. Three elements in a DDE message identify the desired data: *application, topic,* and *item*. Application is the name of the DDE server. Topic refers to a category of data from the server. Item identifies a particular piece of data in the topic.

DDE-aware client A registered software program with a server that understands DDE protocol.

deadband In OptoDisplay, a range of input values for a dynamic attribute that does not affect the appearance of the associated graphic. OptoDisplay uses the deadband value entered for each dynamic attribute. A deadband value of zero allows all input values to impact the dynamic attribute. In order for a graphic to change, a value read must be: a) greater than the deadband plus the last value read, or b) less than the deadband minus the last value read. If the readings fluctuate within a certain bandwidth of the last value read, the graphic display won't change.

delimiter A special character used to separate items written to the same line within a file.

derivative A term used in a *PID loop* calculation. This term acts only on the change in slope of the input signal. Its purpose is to change the output as the input gets near the setpoint to prevent overshooting or undershooting. The derivative term can range between zero and 32,767.

digital Describes data that can assume only two values: on or off, 1 or 0, true or false. Also called discrete.

digital point An input or output point that can have only two values, one or zero (also thought of as on or off, true or false). Push buttons and LEDs are examples of digital devices.

discrete See *digital*.

DLL (Dynamic linking libraries) A type of file that acts as a repository of common routines, that is, shared code any application can access. For example, the file opcproxy.dll from the OPC Foundation is required for communication between OPC client and OPC server applications. Libraries may also be provided in static rather than dynamic form.

DIN rail A standard bracket for mounting hardware. It consists of a single metal strip that can hold one or more devices via screws. Opto 22's products use a 35 mm, symmetric DIN rail.

download The process of copying files from one computer or processor to another, over a network or via a modem. In OptoControl, the process by which strategy information is transferred to a controller.

draw window In OptoDisplay, any window that allows graphics to be placed within it. A draw window has static attributes of position, size, and color. Its dynamic attribute is its draw window visual state.

draw window visual state In OptoDisplay, draw windows have visual states of open, closed, or iconified. An open window scans and updates objects associated with it. A closed window scans only its trends, if it was configured to do so, and does no updating. An iconified window scans all its objects but does no graphic updating.

driver A software program that provides instructions for transferring data between an application program and a peripheral device.

dynamic attribute Attributes of a dynamic graphic object that change when a project is run in OptoDisplay Runtime. Examples of dynamic attributes include color, size, position, and visibility. Dynamic attribute behavior is controlled from sampled tag values, such as controller status, that are set up during configuration.

dynamic link In OptoDisplay, the relationship established during configuration between a tag and a dynamic object such as a graphic, historic log, or trend.

dynamic objects Any configurable object that has dynamic attributes associated with it. Examples of dynamic objects include graphics, historic logs, window managers, and trends. During OptoDisplay Runtime, dynamic objects are updated with values from the scanned controllers, user interaction, or system events.

EEPROM (Electrically-erasable programmable read-only memory) See *flash EEPROM*.

Ethernet A local-area network (LAN) protocol.

Ethernet handler task In OptoControl, an invisible “chart” for handling Ethernet communication. If an M4SENET-100 adapter card is installed in the controller, an *Ethernet handler task* will always be in the task queue.

Ethernet I/O A system used to communicate to field devices (such as meters, valves, and switches) from a local or remote location over an Ethernet network. See *SNAP Ethernet I/O*.

event Incidents or occurrences of significance to the OptoDisplay operator; not always an indication of abnormal conditions. Event examples include disk full, communication problems, and so on.

event log A list of messages for all events that have occurred while a project is being executed by OptoDisplay Runtime. The list can be viewed using the Event Log Viewer by selecting View?Event Log from Runtime. Optionally, at configuration time, the event log can be configured to write the event messages to a file in addition to adding them to its list. Only one event log exists per project. An event log file extension is .msg.

event/reaction (E/R) An I/O element that consists of an event and a corresponding reaction. Each time the event occurs, its corresponding reaction is executed once. An example would be to turn on an output based on an output counter reaching a certain value. OptoControl accepts up to 256 event/reactions on each I/O unit. E/Rs are processed at the brain level. Many Opto 22 brains support E/Rs; check the brain’s data sheet for information.

external value (XVAL) The actual physical value of an input or output point. Depending on whether the I/O point is enabled, this value may or may not correspond to the *internal value* (IVAL) represented in the strategy running on the controller. Used in OptoControl.

fiber optic *See optic*.

firmware Any semi-permanent software that is stored in the non-volatile memory of a hardware device. Sometimes called a kernel. Firmware often functions like an operating system. Brain firmware gives the brain the ability to perform needed functions. For example, it allows a multifunction I/O unit to measure frequency, calculate a PID loop, or communicate with a controller or host computer. Controller firmware is stored in the controller’s EEPROM, usually *flash EEPROM*.

flash EEPROM A type of read-only memory that can be reprogrammed as needed to update the software. Sometimes called simply flash or EEPROM.

flat-ribbon cable See *ribbon cable*.

float A numeric value that includes a fractional component, and thus contains a decimal point. Examples are 3.14159, 1.0, and 1234.2. OptoControl uses IEEE single-precision floats with rounding errors of no more than one part per million. Float values can range from ±3.402824 x 10-38 to ±3.402824 x 1,038. Also called floating-point value.

flowchart See *chart*.

flow-through logic In OptoControl, program logic in which a chart’s instructions are executed in sequential order from beginning to end. The end of the chart is an action or condition block that has no exit. Charts that must include flow-through logic are the Powerup and Interrupt charts. Subroutines must also use this type of logic. Compare to *loop logic*.

freshness A method of improving system performance. When an I/O point is accessed by OptoDisplay, a time value associated with the I/O point is tested. If the tested time value is within the specified freshness, the controller will use its internal value. In this manner, the controller saves the communications overhead needed to get the value from the I/O unit. The freshness value in OptoDisplay is configured as part of a refresh time group. For OptoDisplay and OptoServer, the possible configured freshness rate value is from 0 to 9,999 with units of milliseconds, seconds, minutes, hours, days, or months. The freshness value must be less than or equal to the scan rate.

freshness rate The age of data a controller has for I/O points. When a controller reads I/O points, the value is time stamped. If the age of data is equal to or less than the configured freshness rate, the value is reported to the OptoDisplay Runtime application. If the freshness value is greater than the configured value, the controller must read the I/O point again before reporting.

The higher the freshness value, the less frequently the controller has to scan I/O for new data. For OptoServer, the possible configured freshness rate value is from 0 to 9,999 with units of milliseconds, seconds, minutes, hours, days, or months. The freshness value must be less than or equal to the scan rate.

full-scale value The highest value specified for an analog input or output module. For a 4–20 mA input, for example, the full-scale value would be 20 mA. Note that inputs can assume values higher than the full-scale value if they feature *over-range capability*.

.gml The file extension of a Cyrano strategy. Cyrano strategies are converted to OptoControl strategies when opened in OptoControl.

G4LC32 The Opto 22 G4LC32 controller.

G4LC32ISA The Opto 22 G4LC32ISA controller.

G4LC32SX The Opto 22 G4LC32SX controller.

gain A term used in a PID loop calculation. It is the “P” in PID, since gain is the inverse of proportional band. Gain acts directly on the change in error since the last scan. The gain term in a PID can range between -32,767 and 32,767 but must not be zero. Higher gain results in increased output change.

graphic object An object such as a circle, a trend, or a bitmap, which is used in OptoDisplay.

graphics In OptoDisplay, objects such as lines, boxes, and circles, used to build a visual representation of a system or process. An operator interface is a combination of graphics that visually simulate a real-world process.

grid A visual array of points arranged in the draw window to facilitate the drawing and alignment of graphics. Grids may be turned on or off. Grids are available in both OptoControl and OptoDisplay.

grid point One of the visual alignment points represented by a single dot. A grid is composed of numerous grid points.

handle A number assigned by *DDE* to a DDE-aware *client*. This number matches the number in the log file and log window. This number is all that is known about a client. Handles are not assigned to OptoDisplay projects.

Also, a number assigned by an operating system or driver to a resource (such as an AC28 card) as a means of identifying it. The operating system or driver may assign sequential numbers to resources in order to keep track of them.

heartbeat An empty data packet sent to OptoDisplay simply to let it know OptoServer is up and running. A heartbeat is sent from OptoServer to OptoDisplay if data is not sent within a heartbeat interval. The heartbeat interval is based on the timeout value. Typically, an acknowledgment is received by OptoServer at the network level that the heartbeat was received by the OptoDisplay client. If an acknowledgment is not received, OptoServer eventually disconnects this client. A heartbeat is also used by OptoDisplay to determine whether OptoServer is running. Missed heartbeats signal OptoDisplay to switch to the backup server node.

historic log A dynamic object configured in OptoDisplay to write a selected set of data out to a file. Data is sampled at predetermined intervals and written out to files in the user-selected directory. Each historic log can have an associated Start Trigger and Stop Trigger to initiate and terminate sampling.

historical trend A graphical representation in OptoDisplay of a tag's past values. This graphical representation is displayed as a simulation of a strip chart recording. Up to 16 lines can be displayed simultaneously. Operators can scroll forward and backward to view scanned values at different times.

HMI (Human-machine interface) A software application, or user interface, that people use to interact with machines.

host task An invisible "chart" whose purpose is to communicate to OptoControl (in Debug mode) or to OptoDisplay. This chart is assigned the first 500-microsecond time slice during strategy execution.

I/O (Input/output) The transfer of data to or from a computer system involving communication devices, operator interfaces, and/or data acquisition and control interfaces.

I/O channel See *point*.

I/O module A device that provides an interface between signals received from "real-world" field devices and the logic signals used in computers and controllers. For example, a thermocouple input module can convert a millivolt signal from a thermocouple into a numeric value that can be interpreted by a host computer. A digital output module can send a signal to turn a field device (such as a motor) on or off. Some I/O modules have one point of I/O. Quad Pak and SNAP digital modules feature four points of I/O, and SNAP analog modules generally have two or four points.

I/O mounting rack A device on which I/O modules and brains or brain boards can be installed.

I/O point See *point*.

ICTD (Integrated circuit temperature detector) A probe whose current output is proportional to absolute temperature. ICTDs are highly repeatable and easy to use because they don’t require resistance-measuring circuitry, high-precision voltage amplifiers, or cold-junction compensation.

IEEE Institute of Electrical and Electronics Engineers.

input point An I/O element (typically a point on a module) that brings information into a controller from a process. Examples of devices wired to input points are buttons, switches, and sensors.

integer A whole number with no fractional part. Examples of integer values are -1, 0, 1, 999, and -456. OptoControl uses 32-bit signed integers that can range from -2,147,483,648 to 2,147,483,647 for most uses; 64-bit integers are available for use with the digital-only SNAP-ENET-D64 Ethernet brain, which addresses 64 digital points on a single mounting rack.

integral A term used in a *PID loop* calculation. This term acts only on the current error and is used to reduce the error to zero. The integral term can range between zero and 32,767. The larger the integral value, the larger the output change.

internal value (IVAL) The software representation of an input or output hardware value. This value is stored in the strategy running on the controller. Depending on whether the I/O point is enabled, this value may or may not correspond to the actual *external value* (XVAL) on the hardware. The IVAL is updated to match the XVAL whenever the controller’s program requires the data. Used in OptoControl.

interrupt A signal sent to a controller by an I/O unit that has just registered an event configured within an *event/reaction*. Only event/reactions configured in OptoControl to enable interrupts will trigger an interrupt. Once an interrupt is registered, the *Interrupt chart* will start running.

Interrupt chart A special chart in OptoControl that remains in a suspended state until an interrupt is registered, at which time it is started. The Interrupt chart is one of two charts included in every strategy. (The other is the *Powerup chart*.)

ioManager A utility program included in FactoryFloor; part of OptoEnetUtilities. ioManager is used to configure, manage, assign IP addresses, and load firmware to SNAP Ethernet-based I/O units and the M4-SENET100 Ethernet adapter card.

ISA (Industry Standard Architecture) A common bus architecture on the motherboard of DOS-based computers.

item One of three elements (application, topic, and item) used by a *DDE*-aware *client* in a DDE message to identify a particular piece of data (such as an integer, a string, an I/O point, or a range of cells in a work sheet) from a *server*. For example, between OptoServer and a controller, an item could be a particular I/O point. Between OptoServer and Excel, an item could be a specific cell from a spreadsheet. The syntax for the item is:

```
ControllerName[RefreshGroupName]:ItemType.ItemName[start-end].Field.BITn
```

item ID One of three elements (server, access path, and item ID) used by an *OPC*-compliant *client* in an OPC message to identify a particular piece of data from a *server*. Similar to a DDE *item*.

IVAL See *internal value*.

Kbps (Kilobits per second) A modem or other connection's speed, measured in the number of bits transferred per second. One kilobit is 1,024 bits.

kernel See *firmware*.

ladder logic A type of graphical programming language used by most PLCs (programmable logic controllers). The name comes from the ladder-like structure of the graphical language. This language was developed as a software representation of hard-wired logic, which was done with physical relays and timers.

latch Program logic that can monitor when a control turns on or off. Latches can keep records of I/O and can be programmed to notify the user of a change.

link library A .lib file used during linking to resolve references to *APIs* in a *DLL*. Also called import library.

literal A fixed numeric or string value. Also called a constant.

loader A small software program built into flash chips which allows the downloading of a new *firmware* file. Unlike the kernel (firmware), the loader is permanently stored in a section of the flash memory that is not erasable. When a controller is first turned on, it can either boot to the loader or to the kernel. You should set the controller to boot to the loader only when you want to download a new firmware file (kernel) to the controller. Once the download has been completed, the controller should be set to boot to kernel again.

local computer In OptoDisplay, the computer on which an OptoDisplay project runs. In addition to running the project, the local computer can be used at the same time to collect or save historical data from a *SuperTrend* object. Also see *remote computer.*

log messages Information reported by OptoServer about its communication transactions. Log messages may be displayed to the OptoServer main log window or to a file.

log window The OptoServer main window where log messages are displayed.

loop logic A type of program logic in which instructions are executed continuously. An OptoControl chart with loop logic can have several paths through which the logic may flow depending on various criteria. Compare to *flow-through logic*.

mask An integer variable or literal with one or more specific bits set. These bits define a set of bits for other actions to work on. The mask may be represented as a decimal, hexadecimal, or binary value.

Mbps (Megabits per second) Refers to the transfer rate of data.

mdslog.log The default filename of the message logging file for OptoServer. When the Logging**?**Log to File option is selected in OptoServer, communication messages are sent to the file as well as to the main

OptoServer display window. The file is found in the OptoServer root directory. The file name and its location may be changed by using the OptoServer command Logging?Select Log File.

microsecond One one-millionth of one second, abbreviation µs.

millisecond One one-thousandth of one second, abbreviation ms or msec.

module See *I/O module*.

mounting rack See *I/O mounting rack*.

multitasking A time-slicing procedure used by OptoControl to allow several charts to run at the same time. The controller contains a multitasking kernel that allows it to run up to 32 tasks simultaneously, including any running host tasks, by assigning each task a 500-microsecond time slice.

mwdriver.dll A dynamic linking library that provides communication to controllers. Mwdriver.dll can handle up to 16 connections. An example of a connection is one in which an application with specific port settings requests a session to OptoServer. This is one connection. If another application requests three sessions to OptoServer, each over a different physical port, this second application uses up three connections to mwdriver.dll.

NetBEUI (NetBIOS Extended User Interface) A superset of *NetBIOS* that includes functions at the network and transport layers.

NetBIOS (Network Basic Input/Output System) A standard for supporting network communication at the session level that is independent of the underlying protocol layers. It serves as an application programming interface for data exchange.

node name Another term used to refer to a computer name on a network.

noise Random or extraneous electrical signals irrelevant to a system's operation. Noise can be produced by such external sources as power lines, generators, motors, transformers, and electrical storms. It can also result from internal sources, such as resistors, capacitors, and semiconductors.

numeric table In OptoDisplay, a dynamic object that shows the contents of up to four OptoControl tables containing numeric data.

ohm Unit of resistance, frequently represented by the Greek letter omega (Ω).

OLE (Object linking and embedding) Based on COM, provides integration among applications, even with diverse types of information. Introduced by Microsoft in 1992, OLE is more flexible, efficient, and robust than DDE.

OPC (OLE for process control) A standard for communication among industrial control and office applications.

OPC servers and clients Software applications that comply with the OPC specification. The same application can be a server and a client at different times or even at the same time. Servers are slaves—applications that provide data or carry out instructions. Clients are applications that request data or give instructions. Clients can request a connection to the server that is temporary (to read or write a value once) or continuous (to update a value when the data changes, called *report by exception*).

The same network can include multiple servers exchanging data with multiple clients. As long as all are OPC-compliant, the servers and clients can be from different vendors.

open protocols Protocols and standards that are not proprietary: that is, any company can make use of the architecture of the standard or protocol. For instance, TCP/IP is an open protocol, since many companies use it in their communications software, while no one company owns it.

operator- driven dynamic attributes Connections made to a dynamic object that change the value or state of a variable based on an operator's action in OptoDisplay.

optic (fiber optic) Method of communicating information. Instead of using metal wires to pass data, fine plastic or glass tubing is used to pass light from an LED at one end to an input/output module at the other end. This method is useful for avoiding *noise* (or interference) common with electric metal wire communication.

OptoControl Opto 22's flowchart-based, Microsoft Windows software package used to program Opto 22 controllers.

OptoControl strategy The control program resident in a controller. The .cdb file associated with this strategy is used by OptoDisplay Configurator and OptoServer to obtain information about the controller as well as tag names for the I/O points and variables. The same OptoControl strategy can reside on more than one controller in a system.

OptoDisplay A Microsoft Windows software package used to develop and run animated graphical representations of data from a control system and an OptoControl strategy. It has two main software applications: OptoDisplay Configurator and OptoDisplay Runtime. Configurator is the development environment used to create the operator interface, while Runtime is used to display and run the operator interface with the Opto 22 control system.

OptoDisplay project The collection of draw windows, graphical elements, attributes, and all elements defining one operator interface designed with OptoDisplay Configurator.

OptoDriver Toolkit An Opto 22 product that provides the tools necessary to build custom PC-based control and SCADA solutions using Opto 22 I/O systems. The OptoDriver Toolkit, a CD-ROM product, allows developers to write Microsoft Windows and DOS software applications that can access the Opto 22 brain, rack, and I/O combination that best fits their application. Applications can be developed using high-level languages, such as Microsoft C++ or Microsoft Visual Basic.

OptoKernel The Opto 22 firmware that must be downloaded to a controller to enable control strategies to execute.

optomds.ini A file created when you exit the OptoServer Administrator for the first time. The optomds.ini file contains controller and scan rate information about the options you entered in the OptoServer

Administrator. Information in this file is read by OptoServer once when the program is initially started. If you make configuration changes in the OptoServer Administrator while OptoServer is running, you must stop and restart the OptoServer program to have these changes take effect. The optomds.ini file is created in the OptoServer root directory.

Optomux A serial (RS-422/485) protocol allowing up to 4,096 I/O points to be connected to a computer used with the B1, B2, and B3000 brain boards.

OptoScript A procedural language similar to C, Pascal, or BASIC, which is used within OptoControl flowcharts to simplify some common programming tasks such as string handling, mathematical computations, and control loops. OptoScript code is contained within OptoScript blocks.

OptoScript blocks Hexagonal chart blocks that contain OptoScript code to be executed in an OptoControl strategy. OptoScript blocks can have many entrances, but only one exit.

OptoServer An Opto 22 server application used to acquire information from any number of networked controllers for use by Opto 22, OPC 1.0-compliant, and DDE-aware applications.

OptoServer Administrator One of three programs that make up the OptoServer software package. Communication parameters such as port type, I/O address, baud rate, and timeouts, as well as scan/refresh times, are configured with the OptoServer Administrator.

OptoSniff An Opto 22 utility application used to display communications between Opto 22 applications and Opto 22 controllers.

OptoTerm An Opto 22 utility application used to check communication between Opto 22 applications and Opto 22 controllers. Also used to update controller firmware and download files to controllers.

OptoVersion An Opto 22 utility application used to check the versions of all Opto 22 software installed on a computer.

output point An I/O element (typically a point on a module) wired to hardware that receives information from the controller to control various components of a process. For example, lights, motors, and valves are devices that may be wired to output points.

over-range capability The ability of an analog input module to register values above the specified full-scale value. For example, a 0–10 VDC input module may actually register voltages up to 11 VDC. Above this value, the signal is considered off-scale high.

overrun Also called a scanner overrun notification. A message indicating that the internal OptoDisplay scanner is unable to scan all of the I/O data requests at the rate configured by the user. An overrun can occur if too many points have been configured at too fast a scan rate. An overrun can also occur if the controller takes too long to provide the data at the configured rate.

Pamux A high-speed, high-density distributed I/O system that accommodates both digital and analog brain boards and I/O modules. Pamux supports up to 32 stations containing up to 512 I/O points. A Pamux bus can extend up to 500 feet from a host computer or other programming device. Pamux is ideal for low-noise environments requiring high speed, such as robotics and numerical control.

PID loop An I/O element used to drive an analog input toward a particular value (called the setpoint) and to keep the input very close to that value. Temperature control is a typical application for a PID. PIDs include a gain term (abbreviated P for proportional, which is the inverse of gain), an integral term (I), and a derivative term (D).

point A single input/output data access location. An I/O point can accept either input data (read from a field device) or output data (to be transmitted to a field device). Some I/O modules have one point of I/O. Quad Pak and SNAP digital modules feature four points of I/O, and SNAP analog modules generally have two or four points. Also called a channel.

pointer A type of data you can store in an OptoControl variable. A pointer does not store a value; it stores the memory address of a variable or another OptoControl item, such as a chart, an I/O point, or a PID loop.

port A communication connection on a computer or controlling device through which a process gains access to a network or bus.

Powerup chart The first chart started in any OptoControl strategy. This is the only chart started automatically at powerup or runtime. The Powerup chart is one of two charts included in every strategy. (The other is the *Interrupt chart*.)

priority In OptoControl, the number of consecutive time slices a task can use. All tasks have a priority of one by default.

project In OptoDisplay, a collection of one or more draw windows containing dynamic objects that represent a control system or process.

Quad Pak module An I/O module with four points of discrete I/O in one package.

radian A natural unit of angular measurement equal to 57.29578 degrees of arc. Note that 2π radians = 360 degrees and $2\pi f$ = angular frequency in radians per second (represented by the Greek letter omega, ω), where f = frequency in Hz.

real-time trend In OptoDisplay, a graphical representation of a tag's value as it changes over time. This graphical representation is displayed as a simulation of a strip chart recording. Up to 16 lines can be displayed simultaneously.

recipe Used to download or upload data to an OptoControl program. Recipes allow an operator to make broad changes to program variables for the purpose of tailoring the OptoControl strategy to specific runs or product types. Data is stored in ASCII files, which can be edited with any text editor or word processor that can save the data in ASCII format. Recipes are uploaded or downloaded to the controller in OptoDisplay by toggling a graphic with a recipe dynamic attribute, or by configuring a trigger-based recipe event.

recipe download A recipe action that causes a recipe file to be sent by the OptoDisplay project to the controller.

recipe format file A file resembling a recipe file which contains the data desired for a recipe upload.

recipe upload A recipe action that causes the controller to send information about a recipe to the OptoDisplay project.

re-enable A communication port property used by the OptoServer node (computer) to specify the time interval to wait before checking for a controller response. This property frees up the computer's CPU for other tasks during this waiting period. The OptoServer default re-enable time is 30 seconds.

refresh time Refers to the *scan time* and *freshness* values for data read by OptoControl or OptoServer from a controller. The scan time is how often the software reads the controller's I/O data. The freshness value is the maximum age the data read from the controller can be.

refresh time group A refresh time group defines a pre-configured refresh rate and freshness time applicable to one or more scanned tags. Up to seven separate refresh time groups can be configured within an OptoDisplay project.

remote computer In OptoDisplay, a computer that is connected over a network to a PC running an OptoDisplay project. If it is running the same OptoDisplay project, the networked computer can be used to save historical data from a *SuperTrend* object. Also see *local computer.*

report by exception The basis by which OptoServer notifies its clients that data has changed. Data is reported only if it has changed from the last time it was reported to the client.

resolution The smallest increment of a signal that can be detected by a system. For example, 12-bit resolution describes data accurate to the 12th bit, which implies a possible change of one part in 4,096 or 0.0244 percent.

retries A communication port property that sets the number of times OptoControl or OptoServer tries to communicate with a controller after its first attempt, before timing out. The range for this parameter is from one to nine. The default is one retry.

ribbon cable A flat cable in which the conductors are arranged side by side. Also called flat-ribbon cable.

rollover period In OptoDisplay, the period of time a log or file receives data before that data is written to a new log or file. Rollover periods may be months, days, or hours. File naming conventions are adjusted accordingly.

ROM (Read-only memory) Non-volatile memory, which means that the information stored there is not erased when power is turned off. Hence, it is often used to store *firmware* that controls a processor at a basic level.

RTD (Resistance temperature detector) A metallic probe used to measure temperature based on its thermal coefficient of resistivity.

running Being executed. Describes the state of an OptoControl chart that's neither stopped nor suspended.

Runtime (OptoDisplay) A program from the OptoDisplay package used to run an OptoDisplay project.

scan group A collection of tags that are scanned at a particular rate. The scan rate defines how often controller variables are scanned to refresh OptoDisplay tags. Every OptoDisplay tag belongs to a scan group. There are seven possible scan groups, with configurable refresh times ranging from milliseconds to months. Communication loads can be optimized by changing the refresh times of a scan group.

scanner overrun notification See *overrun*.

scan time The time interval between controller I/O readings by OptoDisplay or OptoServer. Typical scan times are one second or less for time-critical data, and 60 seconds for data that changes less frequently, such as outdoor air temperature. The scan time does not determine the frequency with which OptoDisplay or OptoServer sends data to clients. Unchanging data since the last scan is not reported to a client. The scan time is from zero to 9,999, with units of time in milliseconds, seconds, minutes, hours, days, or months.

scripting See *OptoScript*.

serial communication module An Opto 22 SNAP I/O module that provides two high-speed, isolated channels of serial data on the same I/O mounting rack with digital and analog modules. Used on a SNAP Ethernet-based I/O unit to provide a convenient connection to printers, chart recorders, barcode readers, and other serial devices at any location, serial communication modules are available in RS-232 and RS-485/422 models.

server An application that supplies data requested by *client* applications that are connected with the server, typically over a local area network. OptoServer is a server that provides data to clients from a controller on the network. Examples of clients include DDE-aware, OPC-compliant, and Opto 22 applications such as OptoDisplay.

Also, one of three elements (server, access path, and item ID) used by an OPC-compliant client in an OPC message to identify a particular piece of data from a server. Similar to DDE *application*.

share Automatically created by OptoServer to allow clients access to OptoServer across a network. Each node has resources (such as disks or files) that can be made available for access by others on the network. Resources that can be shared are file directories or DDE Application/Topic pairs. Shares may be password protected to control reading or writing privileges.

sizing handles Small, solid black boxes that appear around a selected graphic or group of graphics. They can be used to change the size of the selected graphics in OptoControl and OptoDisplay.

SNAP-B3000-ENET brain A compact, flexible, high-performance processor for many applications, including industrial control and data acquisition. The SNAP-B3000-ENET brain is designed to remotely interface with a mix of analog and digital SNAP input/output (I/O) modules, and special-purpose modules such as serial communication and PID modules.

SNAP-ENET- D64 brain A compact, flexible, high-performance processor for many applications, including industrial control and data acquisition. The SNAP-ENET-D64 brain is a digital-only processor that controls up to 64 points of digital I/O on one compact Opto 22 mounting rack.

SNAP-ENET-RTC brain A compact, flexible, high-performance processor for many applications, including industrial control and data acquisition. The SNAP-ENET-RTC brain provides the same functions as a SNAP-B3000-ENET brain, but also contains a real-time clock with battery backup.

SNAP-ENET-S64 brain A low-cost, high-quality Ethernet-based brain for simple monitoring, control, and data acquisition using a wired Ethernet network. Ideal for high-density commercial and industrial applications. The SNAP-ENET-S64 brain offers simple digital and serial capabilities, plus full analog features, on a16-module rack

SNAP Ethernet I/O A compact Opto 22 system consisting of a SNAP Ethernet brain, I/O modules, a mounting rack, and an Ethernet connection. Because SNAP Ethernet I/O uses standard Ethernet TCP/IP technologies, you can monitor and control analog, digital, and serial inputs and outputs anywhere.

SNAP I/O A system made up of compact, modular components that are designed to work together. SNAP components include high-density AC and DC digital I/O modules, intelligent analog I/O modules, I/O mounting racks, and a common I/O processor (brain). Components simply snap in place.

SNAP Simple I/O An Ethernet-based I/O unit consisting of a brain, I/O modules, and a mounting rack. SNAP Simple I/O provides simple digital, serial, and full analog functions.

snap on A characteristic that causes points of a graphic to be placed only on *grid points*. Snap on is used to make the drawing and alignment of graphics easier. The snap option is very useful when object sizes and alignments need to be consistent. Snap on is sometimes referred to as magnetism. Snap on is available in OptoDisplay, but is not the default.

snap off A characteristic that means a graphic will not be placed at the nearest grid point when it is moved in the draw window; it can be placed anywhere. Snap off is the default in OptoDisplay.

SNAP-WLAN-FH-ADS brain A compact, flexible, high-performance processor for many applications, including industrial control and data acquisition. The SNAP-WLAN-FH-ADS brain provides the same functions as a SNAP-B3000-ENET brain, but communicates via wireless LAN.

static objects Graphics in OptoDisplay that do not have a dynamic connection. Changes in a controller do not affect a static object. Static objects are used to make a process graphic more familiar to the operator.

stopped Inactive; not being executed. Describes the state of an OptoControl chart that is neither running nor suspended.

strategy A set of instructions that directs an automation process. Strategies can be designed as one or multiple flowcharts within OptoControl. A strategy is sometimes referred to as a control program or application.

string A sequence of ASCII characters grouped together. These characters can include standard alphanumeric characters as well as control codes and extended characters.

subroutine A chart that can be created and saved independently of a strategy. Subroutines can be included in any number of strategies and can be referenced just like any command in OptoControl.

SuperTrend A rectangular dynamic object in OptoDisplay that graphs the change in a variable or set of variables over time. Trends show variables on the vertical axis and time on the horizontal axis. SuperTrends show real-time, historical, or combined trends, and include 16 trend pens, point markers, and log file rollover. In historical modes, you can scroll through data and see point information. Zooming is preset on the x-axis and variable for each pen on the y-axis. Compare to *basic trend*.

suspended Temporarily paused. Describes the state of an OptoControl chart that is neither running nor stopped.

Symbol Factory In OptoDisplay, a built-in library of graphics designed for industrial automation, which you can use in creating an *HMI*.

table A one-dimensional array representing several numeric, string, or pointer values, each referenced by an index number. A table of length 20 can store 20 elements at indices zero through 19.

tag A symbolic name used to identify a piece of data such as an I/O point, alarm, variable, or system condition. For example, a connection from a graphic in OptoDisplay can be made to an I/O point in an OptoControl strategy. The I/O point is referred to as a "tag" in OptoDisplay.

tag name The name of a piece of data in a controller as configured in an OptoControl strategy.

task See *chart*.

task queue The list of charts (tasks) to run concurrently in the OptoControl strategy. Up to 32 tasks (including the *host task* and the *Ethernet handler task*) can be included in the queue.

TERM1 A terminator board for the Pamux bus. The final Pamux brain board on the bus must be terminated with a TERM1 or TERM2 terminator board.

TERM2 A terminator board for the Pamux bus, identical to the TERM1 in size and function. However, the TERM2 offers lower line impedance than the TERM1. This may prove useful when using a cable that differs from recommended specifications.

thermocouple A temperature sensor that includes a junction of two different metals. Temperature can be derived from the voltage produced at the contact point of the metals.

throughput On a controller, the time its microprocessor takes to complete transactions (commands). Throughput is greatly affected by how fast the controller can read and write I/O from its attached I/O units.

timeout Sets the time between each communication attempt from OptoControl or OptoServer to a controller. The number of communication retries after the initial attempt is set by the *retries* parameter.

timer In OptoControl, a numeric quantity representing elapsed time in units of seconds with resolution in milliseconds. Down timers continuously count down to zero, and up timers continuously count up from zero. Timer range is from 0.001 to 4.611686 x 10^{15} seconds.

time slice A fixed unit of CPU time. OptoControl uses a time slice of 500 microseconds (one-half millisecond).

toolbox A set of drawing tools used to create graphics in OptoDisplay. The toolbox may be moved anywhere in the drawing area to facilitate drawing.

topic Identifies a category of data from OptoServer. It is one of three elements (application, topic, and item) used by a *client* in a *DDE* message to identify a particular piece of data (such as an integer, a string, an I/O point, a range of cells in a work sheet, etc.) from a *server*. For OptoServer, the topic is generally OPTO_MDS_0. For Excel, the topic corresponds to the spreadsheet's file name.

trend See *basic trend*, *SuperTrend*, and *XY plot*.

trend pen A dynamic object used by trends in OptoDisplay. The value of its tag provides the data for the line graphs in the trends.

trigger In OptoDisplay, a trigger is configured with a tag and a condition to which that tag is compared. The condition comprises a value and a mathematical relationship. The specified tag is sampled and compared against this specified condition. Triggers are edge sensitive and only activate with a positive transition from a non-triggered state. Triggers are often used to start and stop sampling of other tags associated with dynamic attributes of a dynamic object in OptoDisplay.

UCA4 A general-purpose adapter card used to connect any TTL device to the Pamux bus. Its purpose is to allow a user to build a custom interface to a Pamux system.

under-range capability The ability of an analog input module to register values below the specified zero-scale value. For example, a 0–10 VDC input module may actually register voltages down to -0.25 VDC. Below this value, the signal is considered off-scale low.

variable A quantity that can assume any of a set of values in OptoControl. The name of a variable remains fixed during strategy execution, but its value can change. There are six types of variables: numeric, string, pointer, numeric table, string table, and pointer table. The difference between them is the type of data they store.

visual state In OptoDisplay, the appearance of a window. It can be open, closed, or iconified. An open draw window requires regular scanning of a controller to update its tag-connected graphics. An iconified window scans a controller for new data, but doesn't update its graphics. A closed window optionally scans any trends it may have, but does not update displays.

volts Measure of electrical potential, abbreviated V. Voltage is always expressed as the potential difference in available energy between two points. One volt is the force required to produce a current of one ampere through a resistance or impedance of one ohm.

watchdog A fail-safe mechanism that can be used to set I/O points to a "safe" state if communication with the controller is lost. A watchdog timeout can be specified for an I/O unit. If no communications are received from the controller within this time span, then the output points on the I/O unit will be set to their configured watchdog state. Watchdogs are configured in OptoControl.

watch window In OptoControl, a window you can create to monitor one or more strategy elements (I/O units, digital and analog points, event/reactions, variables, and so on) while debugging the strategy.

window manager In OptoDisplay, a draw window combined with a trigger allowing dynamic control of another draw window's visual state. Specifically, it is a dynamic object used to activate, deactivate, or iconify one or more windows based on changes in process variables or operator actions. Each window manager has an associated trigger and list of draw windows and their visual state transitions. Up to 1000 window managers per project are allowed.

.wth File extension for a *watch window* file.

XVAL See *external value*.

XY plot A rectangular dynamic object in OptoDisplay that graphs data from two numeric *tables* in an OptoControl strategy. The XY plot uses the numeric data in one table for x-axis values and data in the other table for y-axis values. Compare with *basic trend* and *SuperTrend*.

zero-scale value The lowest value specified for an analog input or output module. For a 4–20 mA input, for example, the zero-scale value would be 4 mA. Note that inputs can assume values lower than the zero-scale value if they feature *under-range capability*.

OptoServer Index