

# OPTOTERMINAL QUICK START GUIDE

Form 1338-070321—March, 2007

## **OPTO 22**

---

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](http://www.opto22.com)

### **Product Support Services**

800-TEK-OPTO (835-6786) or 951-695-3080

Fax: 951-695-3017

Email: [support@opto22.com](mailto:support@opto22.com)

Web: [support.opto22.com](http://support.opto22.com)

OptoTerminal Quick Start Guide  
Form 1338-070321—March, 2007

Copyright © 2002–2007 Opto 22.  
All rights reserved.  
Printed in the United States of America.

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.

---

Cyrano, Opto 22 FactoryFloor, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, mistic, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDataLink, OptoDisplay, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, PAC Control, PAC Display, PAC Manager, PAC Project, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC System, SNAP Simple I/O, SNAP Ultimate I/O, and SNAP Wireless LAN I/O are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C+++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Unicenter is a registered trademark of Computer Associates International, Inc. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric. Wiegand is a registered trademark of Sensor Engineering Corporation. Nokia, Nokia M2M Platform, Nokia M2M Gateway Software, and Nokia 31 GSM Connectivity Terminal are trademarks or registered trademarks of Nokia Corporation. Sony is a trademark of Sony Corporation. Ericsson is a trademark of Telefonaktiebolaget LM Ericsson.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

# Table Of Contents

OPTO 22

<b>Chapter 1: Installation</b> .....	<b>1</b>
Introduction .....	1
About Opto 22 OptoTerminals .....	1
OptoTerminal-G70 and OptoTerminal-G75 .....	2
Opto 22 Products Compatible with OptoTerminals .....	2
Related Documentation .....	2
For Help .....	3
Hardware and Software Setup .....	3
What You Will Need .....	3
Simulation Mode in Qlarity Foundry Software .....	3
Power .....	4
Using an OptoTermCable .....	4
Configuring an OptoTerminal .....	4
Entering Ethernet Network Settings .....	5
Selecting Develop or Download Mode .....	5
Saving Configuration Settings .....	6
Installing Qlarity Foundry Software .....	6
<b>Chapter 2: Quick Start</b> .....	<b>7</b>
Choosing Your Quick Start Section .....	7
Quick Start for Systems Running PAC Control .....	8
Hardware and Software Requirements .....	8
Preparing the SNAP PAC .....	9
Creating the Qlarity Foundry Application .....	10
Configuring the OptoTerminal for Downloading .....	17
Downloading and Running the Application .....	18
What's Next? .....	19

---

Quick Start for I/O Units .....	20
Hardware and Software Requirements .....	20
Preparing the I/O Unit .....	20
Creating the Qlarity Foundry Application .....	22
Displaying a Value from the Memory Map .....	24
Adding a Button .....	27
Changing the Display Based on the Button .....	28
Using a Slider to Change the Display .....	32
Working with a Container Object .....	33
Configuring the OptoTerminal for Downloading .....	36
Downloading and Running the Application .....	37
What's Next? .....	38

### **Chapter 3: Using Opto 22 Library Objects ..... 39**

Introduction .....	39
Opto 22 Communications Objects .....	39
Opto 22 Object Libraries .....	40
Choosing a Library .....	40
If You Need Both Libraries .....	40
Common Uses for Opto 22 Library Objects .....	41
Opto 22 PAC Control Library Object Reference .....	41
Opto 22 Memory Map Library Object Reference .....	43

### **Chapter 4: Troubleshooting ..... 45**

Solutions to Common Problems .....	45
Updating OptoTerminal Firmware .....	46
Reviewing Diagnostic Information .....	46

# Installation

## Introduction

This guide shows you how to configure an Opto 22 OptoTerminal™ operator display terminal and connect it to an Opto 22 SNAP PAC™ controller over a standard 10/100 Mbps Ethernet network.<sup>1</sup>

The guide also includes basic instructions for using the included Qlarity™ Foundry™ programming software to communicate with these Opto 22 systems.

Additional information, including detailed configuration and programming instructions, is provided in the documents included with the OptoTerminal.



**OptoTerminal-G75 (left) and OptoTerminal-G70 (right)**

**NOTE:** While Opto 22 OptoTerminals are based on QSI Corporation's QTERM™ -G70 and QTERM-G75 terminals, these QSI terminals cannot communicate with Opto 22 systems.

## About Opto 22 OptoTerminals

Opto 22's OptoTerminal family brings a programmable, Ethernet-enabled, graphics-based operator interface to any location with Opto 22 SNAP Ethernet-based I/O systems. OptoTerminal mountable display terminals are used by operators to send commands to and receive real-time data from plant floor equipment and other devices connected to Opto 22 hardware. Ideal for automation, control, data acquisition, and remote monitoring, an OptoTerminal offers a compact and durable operator interface for the plant floor and remote locations.

<sup>1</sup> Legacy hardware SNAP Ultimate I/O™, SNAP Ethernet I/O™, SNAP Simple I/O™, SNAP-IT, or a system using the SNAP-LCE industrial controller may be used as well. However, this manual will only address systems using SNAP PAC hardware.

## OptoTerminal-G70 and OptoTerminal-G75

Two OptoTerminal models are currently available from Opto 22: the OptoTerminal-G70 and the OptoTerminal-G75. The primary difference between these two models is the size of the screen, and the instructions in this guide apply to either terminal. For detailed information and technical specifications for each OptoTerminal, see Opto 22 form 1340, the *OptoTerminal Data Sheet*, and Chapter 1 and Chapter 2 of Opto 22 form 1343, the *OptoTerminal Hardware User's Manual*.

## Opto 22 Products Compatible with OptoTerminals

An OptoTerminal can control and monitor the following Opto 22 hardware:

SNAP PAC	SNAP Ultimate I/O	SNAP Ethernet I/O	SNAP Simple I/O	Other Hardware
SNAP-PAC-S1	SNAP-UP1-ADS	SNAP-B3000-ENET	SNAP-ENET-S64	SNAP-LCE industrial controller
SNAP-PAC-R1	SNAP-UP1-D64	SNAP-B3000-ENET-RTC		Any SNAP-IT packaged unit
SNAP-PAC-R2	SNAP-UP1-M64	SNAP-ENET-D64		
SNAP-PAC-EB1		SNAP-WLAN-FH-ADS		
SNAP-PAC-EB2				

In this document, we will use an OptoTerminal to communicate with a SNAP-PAC-R1 controller.

## Related Documentation

You'll also need some of the following documents for the Opto 22 hardware your OptoTerminal will communicate with and for the PAC Control™ software if you will be using a strategy:

To use this	See this	Opto 22 form #
SNAP PAC S-Series Controller SNAP PAC R-Series Controller	<i>SNAP PAC S-Series Controllers User's Guide</i> <i>SNAP PAC R-Series Controllers User's Guide</i>	1592 1595
PAC Control strategy running on a SNAP PAC controller	<i>PAC Control User's Guide</i>	1700

Some of these documents are on the CD that came with the SNAP PAC controller, and all forms are available on the Opto 22 Web site ([www.opto22.com](http://www.opto22.com)) for downloading. The easiest way to find one is to search for the form number.

## For Help

If you have problems connecting, configuring, or using an OptoTerminal and cannot find the help you need in this guide or on the Opto 22 Web site, contact Opto 22 Product Support.

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

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

## Hardware and Software Setup

### What You Will Need

You'll need the following items to connect, configure, and program an OptoTerminal:

- P400 or faster personal computer running Microsoft® Windows® with a 10/100 Mbps Ethernet adapter card. The adapter card should have a valid IP address on the same network subnet that the OptoTerminal is connected to.
- Compatible Opto 22 hardware connected to an Ethernet network. This hardware will typically include a SNAP PAC controller, one or more SNAP PAC racks, and SNAP I/O modules required for your application.
- OptoTerminal-G70 or OptoTerminal-G75 with firmware version 2.50 or newer. To determine your firmware version or upgrade it, see instructions in the *OptoTerminal Qlarity Foundry User's Manual*. Current firmware is available for download from our Web site at [www.opto22.com](http://www.opto22.com).
- Ethernet crossover cable (for direct connection from the OptoTerminal to the PC), or an available connection to a standard 10BASE-T or 100BASE-TX Ethernet network.
- 10–26 VDC power source, connected through the terminal's DB9 female serial connector. See ["Power"](#) in the following section for important additional information about power requirements.

### Simulation Mode in Qlarity Foundry Software

If your OptoTerminal is not yet installed, but you want to complete the quick start exercises in this guide now, you can simulate an OptoTerminal by using the included Windows-based Qlarity Foundry software running on a PC. Skip to ["Installing Qlarity Foundry Software"](#) on page 6.

## Power

Power is supplied to an OptoTerminal through the unit's serial connector. To provide this power, a serial connector must be wired to a power supply at the installation site. See the *OptoTerminal Hardware User's Manual* for connector pinouts and specific power requirements for each OptoTerminal model.

**IMPORTANT: An OptoTerminal does not come with a power cable.**

Opto 22 OptoTerminals come without an included power cable because power can be supplied in many different ways at a final installation site based on the type of application, equipment used, environment, and other factors. Typically you will build a power cable for the OptoTerminal to match installation site requirements, although some applications may be able to use the optional OptoTermCable, described below.

## Using an OptoTermCable

An optional OptoTermCable is a combination serial cable and power supply that provides power for an OptoTerminal-G70 or OptoTerminal-G75. Two versions of the OptoTermCable are available to meet the power requirements of each OptoTerminal model.

An OptoTermCable can sometimes be used at an installation site, but it is typically used at a location such as an office where OptoTerminal applications are developed before deployment. An OptoTermCable includes a serial cable for connecting to a PC; this serial connection is useful for receiving diagnostic messages while programming a terminal. Contact Opto 22 for more information on the OptoTermCable.

## Configuring an OptoTerminal

Before programming an OptoTerminal with Qlarity Foundry software, you'll need to configure the terminal's built-in settings and connect it to an Ethernet network. Remove the unit from its packaging and then perform the following basic steps to do this. For detailed information on setting up and installing the terminal, see Opto 22 form 1343, the *OptoTerminal Hardware User's Manual*.

1. Plug the Ethernet cable into the RJ-45 connector on the back of the unit.
2. Press and hold a finger or thumb on the upper left-hand corner of the screen as shown below.



3. While holding your finger on the screen, connect power to the terminal. Release your finger after the Power On Setup screen appears.

### Entering Ethernet Network Settings

To communicate on an Ethernet network, an OptoTerminal must have an IP address and subnet mask assigned to it. A gateway address may also be required. Contact your network administrator for this information.

1. Press the Back or Next buttons at the bottom of the screen until NETWORK is highlighted, and then press Select.  
IP is highlighted.
2. To enter an IP address, press Select to highlight a digit in the IP address, and then press Back or Next to choose a number.
3. To enter a subnet mask, press Back or Next to highlight Sub, and then press Select. Change the subnet mask digits as described in [step 2](#) above.
4. To enter a gateway address, press Back or Next to highlight Gate, and then press Select. Change the gateway address digits the same way.

### Selecting Develop or Download Mode

To load an application on the OptoTerminal, you must first set the terminal to one of two modes: *Develop* or *Download*.

- **Develop mode** saves an application to the terminal's volatile (non-permanent) memory only, meaning that if power to the terminal goes off, the downloaded application will be erased. Because Develop mode does not load an application into the terminal's non-volatile flash memory, the application runs almost immediately after being loaded. This mode is best suited for programming and testing an application on the OptoTerminal.
- **Download mode** saves an application into the terminal's flash memory. This means that if power to the terminal goes off, the downloaded application will remain in memory and restart the next time the OptoTerminal is turned on. The primary drawback to Download mode is that the application requires a longer time to start up after being loaded to the terminal. Download mode is normally used when the OptoTerminal is deployed at the application site.

To set the OptoTerminal to Develop or Download mode, do the following:

1. Press the Back or Next buttons at the bottom of the screen until FLASH MEMORY is highlighted, and then press Select.  
App Mode is highlighted. (If your menu is different, your firmware may be older than version 2.50. Follow instructions in the *OptoTerminal Qlarity Foundry User's Manual* to install new firmware on the OptoTerminal.)
2. Press Select.  
Run App is highlighted.
3. Press Next to change Run App to either Develop or Download and then press Select.  
App Mode is highlighted.

### **Saving Configuration Settings**

- 1.** Press Back or Next until DONE is highlighted, and then press Select.  
Save and Exit is highlighted.
- 2.** Press Select to save your configuration changes.  
The terminal is reset. You can now download to the terminal an application created using Qlarity Foundry software.

### **Installing Qlarity Foundry Software**

In your CD-ROM drive, insert the OptoTerminal CD. The installation wizard should start automatically. If it doesn't, start Windows Explorer, navigate to your CD-ROM drive, and double-click Setup.exe to begin installation.

To start Qlarity Foundry, click the Windows Start button and select Programs→Opto 22→Qlarity Foundry for Opto 22→Qlarity Foundry.

# Quick Start

## Choosing Your Quick Start Section

The quick start sections in this guide step you through using Qlarity Foundry software to create an OptoTerminal application for use with Opto 22 hardware. OptoTerminal can read and write two different kinds of data to Opto 22 Ethernet-based hardware:

- Data in a PAC Control strategy running on a SNAP PAC controller.<sup>1</sup>
- Data in the SNAP PAC's Ethernet-based I/O unit's memory map.<sup>2</sup>

Since the design of the Qlarity Foundry application you develop depends on which data you need to use, this chapter contains two sets of quick start instructions. Choose your quick start instructions based on the following table:

Are you running a PAC Control strategy?	Use this quick start section
Yes	<a href="#">"Quick Start for Systems Running PAC Control" on page 8</a>
No	<a href="#">"Quick Start for I/O Units" on page 20</a>

*NOTE: It is possible that you might be running a PAC Control strategy and still need data from the memory map in a SNAP PAC I/O unit. If so, you may also want to read the Quick Start steps on [page 20](#).*

---

<sup>1</sup>Other Opto 22 hardware can communicate with an OptoTerminal using data from an ioControl or PAC Control strategy. However, in this manual, we will only reference a SNAP PAC.

<sup>2</sup>Other Opto 22 hardware can communicate with an OptoTerminal using data from a memory map. However, in this manual, we will only reference a SNAP PAC.

## Quick Start for Systems Running PAC Control

This section steps you through using Qlarity Foundry software to create an OptoTerminal application for use with a SNAP PAC that is running a PAC Control strategy. Used this way, the OptoTerminal reads and writes data to the PAC Control strategy. In this quick start, you'll do the following:

- Open a sample PAC Control strategy and download it to the controller or I/O unit.
- Create a new Qlarity Foundry workspace and add objects from the PAC Control object library.
- Save your Qlarity Foundry application and download it to the OptoTerminal.

### Hardware and Software Requirements

To complete this quick start, you need the following:

- **SNAP PAC R-Series Controller and SNAP rack**
  - Mount the SNAP PAC controller on the appropriate SNAP PAC rack and connect it to your computer following the instructions in Opto 22 form 1595, the *SNAP PAC R-Series Controller User's Guide*.
  - Assign an IP address to the SNAP PAC R controller following the instructions in Opto 22 form 1704, the *PAC Manager User's Guide*.

*NOTE: For this quick start, actual I/O points aren't used and SNAP I/O modules aren't needed. Instead, a sample PAC Control strategy running on the SNAP PAC R-series controller calculates internal values (or IVALs).*

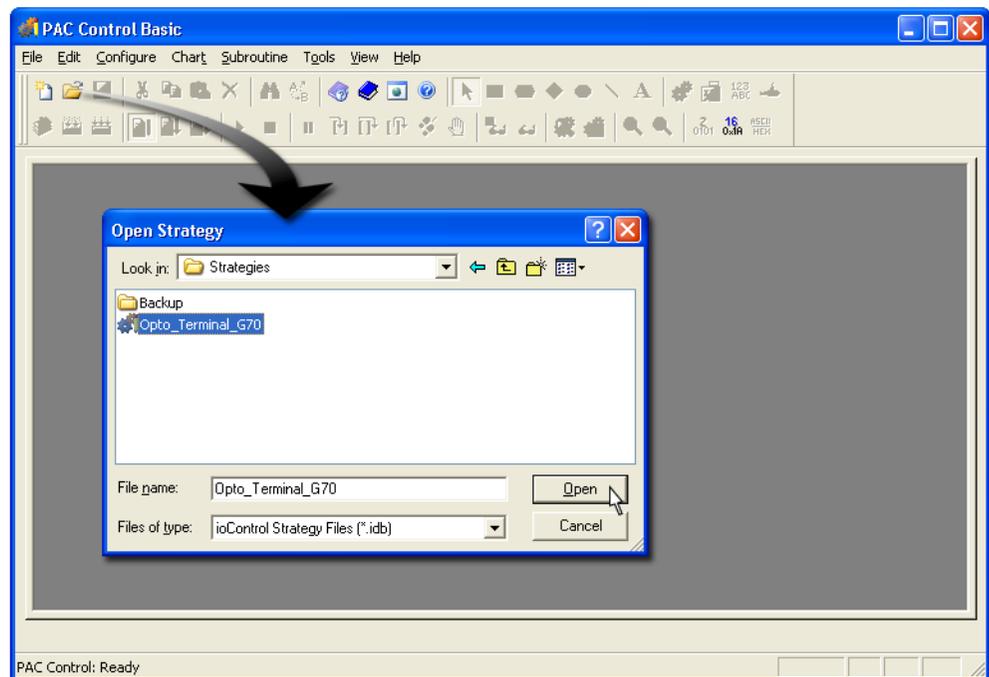
- **OptoTerminal-G70 or OptoTerminal-G75**—If you haven't already set up your OptoTerminal and connected it to an Ethernet network, see ["Configuring an OptoTerminal"](#) on [page 4](#) for instructions.

*NOTE: If you don't have an OptoTerminal available, you can still complete this quick start using Simulation mode in Qlarity Foundry software.*

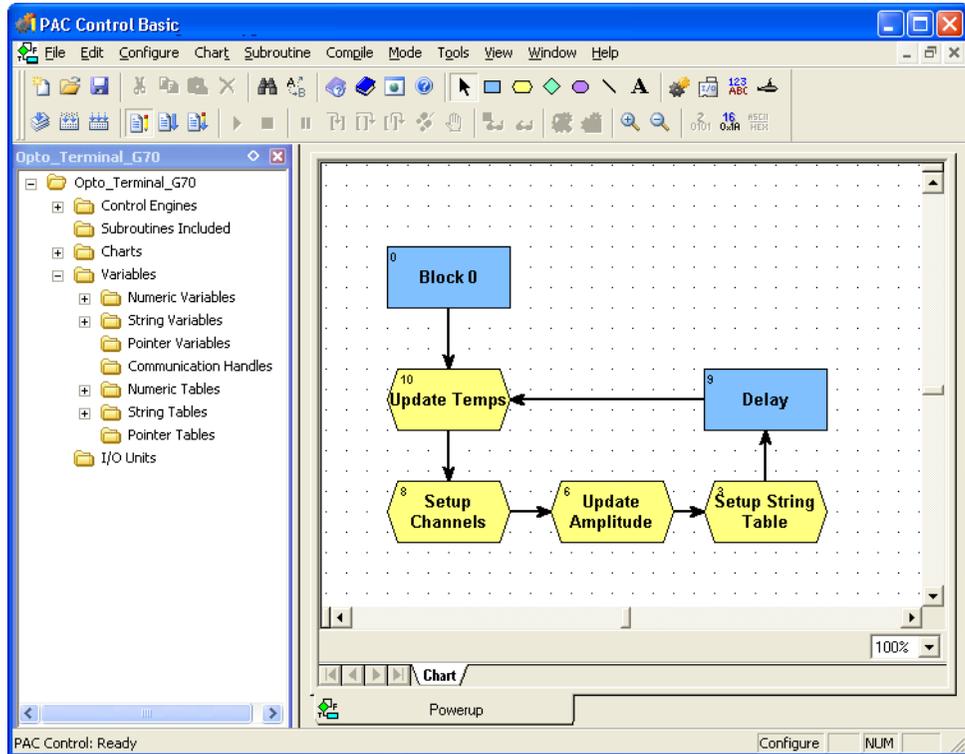
- **PAC Control software version 8.0 or newer**—See Opto 22 form 1700, the *PAC Control User's Guide*, for installation instructions and information on creating a PAC Control strategy. For this quick start you will use a sample strategy, which is provided.
- **Opto 22 PAC Manager™ software version 8.0 or newer**—PAC Manager software is included on the CD shipped with SNAP PAC controllers. You can also download the latest version of PAC Manager from the Opto 22 Web site at [www.opto22.com](http://www.opto22.com).
- **Qlarity Foundry software version 2.50 or newer**—If you haven't already installed Qlarity Foundry software on your computer, see ["Installing Qlarity Foundry Software"](#) on [page 6](#) for instructions.

## Preparing the SNAP PAC

1. From the Windows Start menu, choose Programs→Opto 22→PAC Project Software 8.0→PAC Control Basic.
2. In the PAC Control window that opens, choose File→Open Strategy, browse to the directory C:\Program Files\Opto22\QlarityFoundry\Samples\Strategies, select the file Opto\_Terminal\_G70.idb, and then click Open.



The PAC Control strategy opens in PAC Control Basic. Your PAC Control Basic window should look like this:



3. In the Strategy Tree at the left, click the plus sign next to Control Engines. Double-click the SNAP PAC R-series controller listed there and change it to specify the SNAP PAC you're using for this exercise.

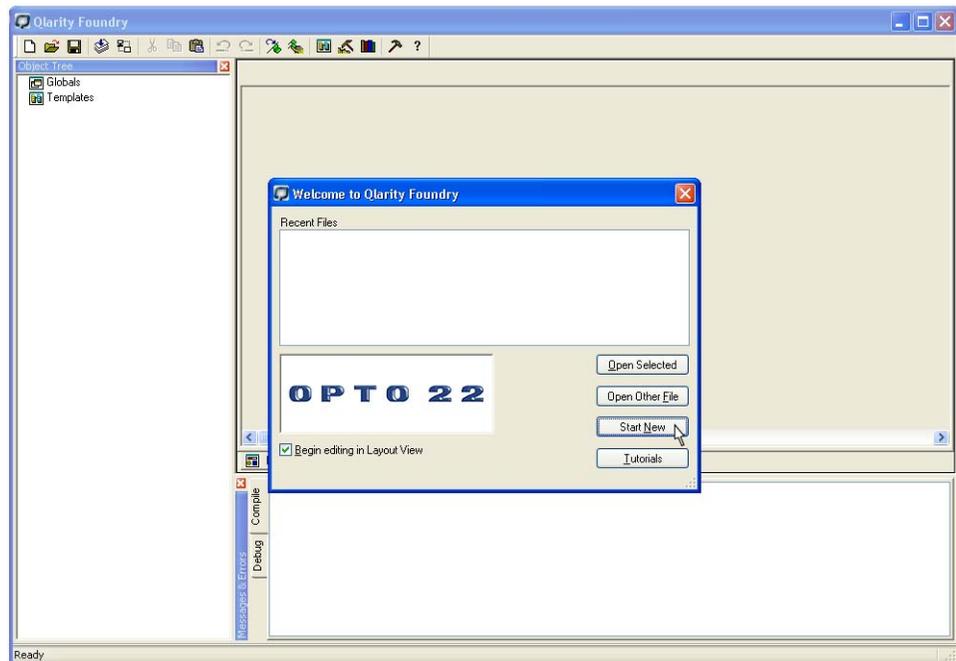
Refer to the *PAC Control User's Guide* if you need help.

4. Choose Mode→Debug to download the strategy to the controller.
5. If prompted to save the strategy before starting the debugger, click Yes.
6. If the message "Powerup Clear Expected from Controller" appears, click OK.
7. If the Download Warning dialog box appears, click Yes.
8. Choose Debug→Run to start the strategy running on the controller.
9. From the File menu, select Exit to close PAC Control.

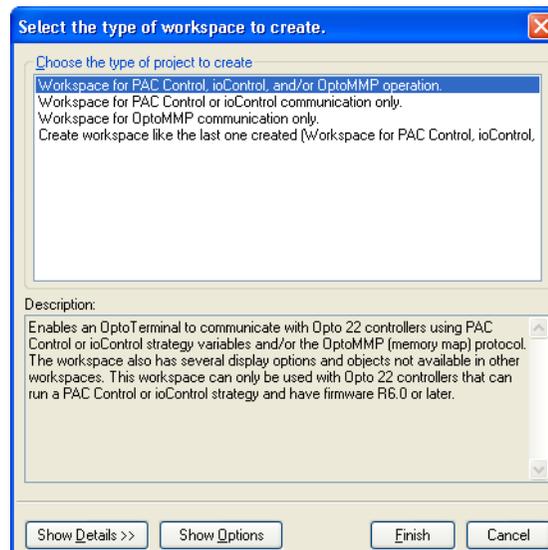
## Creating the Qlarity Foundry Application

1. From the Windows Start menu, choose Programs→Opto 22→Qlarity Foundry for Opto 22 Version 2.50→Qlarity Foundry.

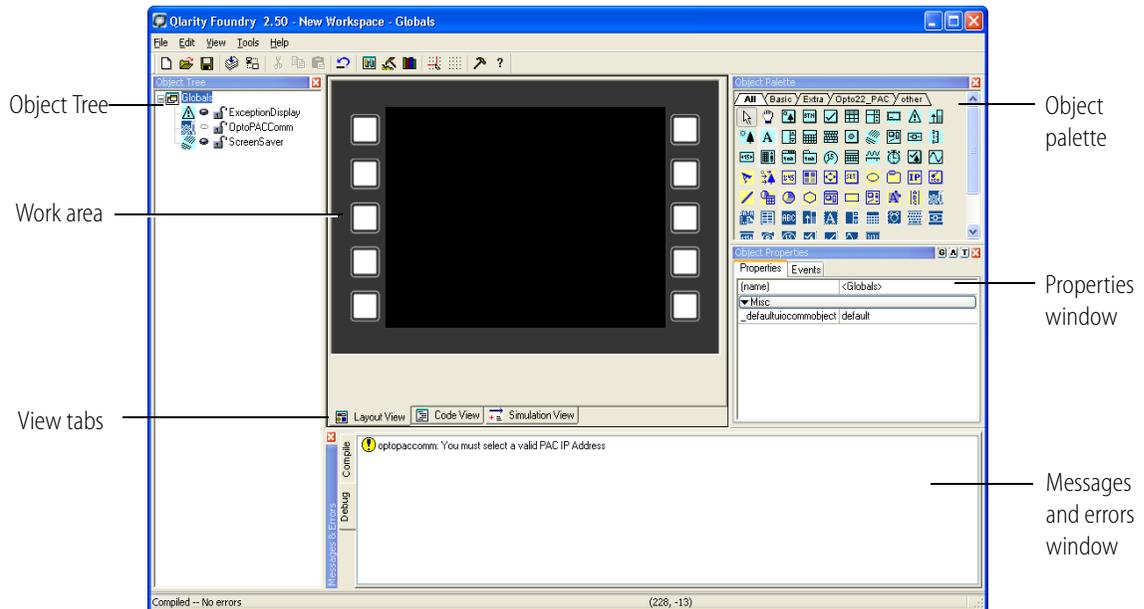
The Qlarity Foundry main window opens, along with the Welcome dialog box.



2. Click Start New.



3. Click to highlight "Workspace for PAC Control and/or OptoMMP operation."
4. Click Next and notice the options available, but don't check any of them now.  
For more information on these options, see the *OptoTerminal Qlarity Foundry User's Manual*.  
The Qlarity workspace opens.



5. Notice the main elements in the workspace:

**Object Tree**—Similar to the Strategy Tree in PAC Control. Any object you create in Qlarity Foundry is listed here. Some objects may be automatically added to a workspace when it is created, including the following:

- ExceptionDisplay—displays unhandled Qlarity exceptions (errors)
- OptoPACComm—handles the communication between the computer and/or terminal and the controller or I/O unit.

**Work area**—This area looks like the OptoTerminal display. Similar to a drawing window in Opto 22's PAC Display™ human-machine interface (HMI) software, you place objects here in the work area as you create your application.

**View tabs**—Click a tab to change the view of the work area.

- Layout View (shown) is a graphics mode used to lay out the display. Generally, unless you are making changes to the programming code, you will work in Layout View.
- Code View is used to write or edit an object's programming code. Typically you use this mode only if you are customizing objects or creating your own objects.
- Simulation mode lets you connect to your controller or I/O unit and test your display without needing to have an OptoTerminal.

**Object palette**—Shows icons for all available objects. Similar to the Toolbox in PAC Display. To add an object, first click its icon, then draw in the work area. To see an icon's name and description, move the mouse over it. Click a tab to see icons in an individual library, or click the All tab to see all available icons.

**Properties window**—When you click an object in the Object Tree or in the work area, its properties are listed here. You can change a property by clicking on its value and typing in a

new value or choosing a value from a drop-down list. To see a property's description, move the mouse over its name.

**Messages and errors window**—Displays errors or other messages generated when the workspace is compiled or during a debug operation. These messages can help you identify and correct problems with the workspace.

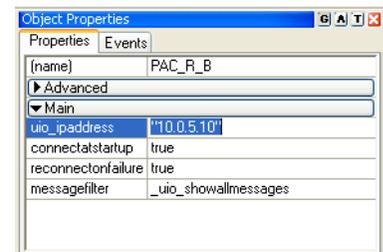
6. In the Object Tree, click the OptoPACComm object to set communication properties. Notice that the list in the Properties window changes.
7. In the Properties window, with OptoPACComm selected, click the value next to the (name) property. Type in a name for the communication object (normally the name of the controller or I/O unit the communications object represents).

In this example, we've used PAC\_R\_B as the name of the communication object.

8. Now click the value next to the property ipaddress and enter the IP address of the controller you're using for this exercise, as shown at right.

You don't need to enter the quotation marks; they are automatically added for you.

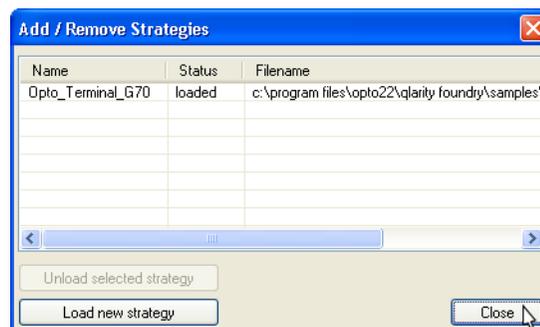
*NOTE: You can add additional OptoPACComm objects if your OptoTerminal needs to communicate with additional controllers or I/O units.*



9. Choose Tools→Select Strategy.
10. In the Open dialog box, navigate to the directory C:\Opto22\Qlarity Foundry\Samples\Strategies, select the file Opto\_Terminal\_G70.idb.txt, and then click Open.

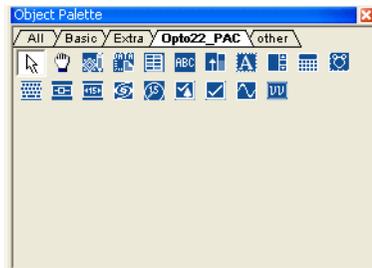
Notice that the file you choose is the **.idb.txt** file, not the .idb file. The .idb.txt file is automatically created for a strategy when the strategy is saved in PAC Control.

The Add/Remove Strategies dialog box appears, showing the sample strategy:

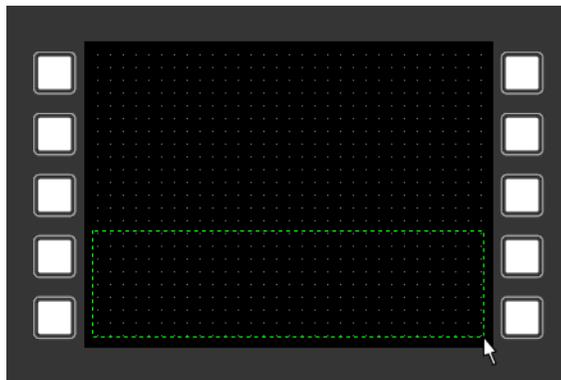


For now, we'll use the sample strategy; but you can add your own strategies in the same way to access the tag database and read and write data to your strategy variables. You can also add more strategies if needed, for example if your Qlarity Foundry application will monitor multiple controllers or I/O units running different strategies.

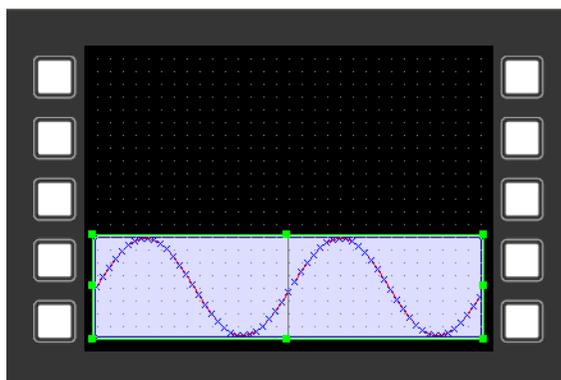
11. Confirm that the correct PAC Control strategy (or strategies) have been selected, and then click Close.
12. Now choose File→Save to save your work to a project file. In the Save As dialog box, give the file a name, navigate to where you want it saved, and then click Save.
13. In the Object palette, click the OptoPAC tab to see the OptoPAC library objects that can be used in your application. (Other objects are hidden; you can see them again by clicking other tabs.)



14. Click the PACTrendChartV2 object . Move the cursor to the work area, and then click and drag the mouse to draw a rectangle similar to the one shown below.



15. Release the mouse button.  
The TrendChart object now has solid lines and a graph, as shown in the following example.

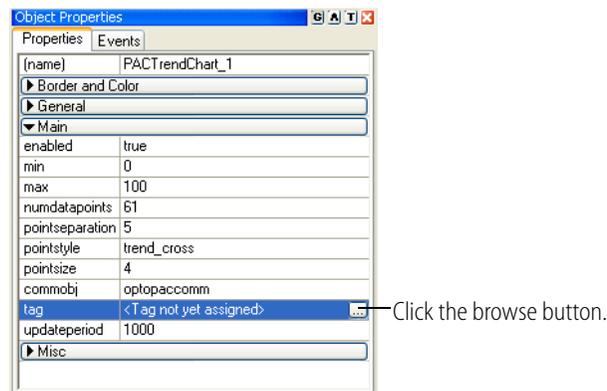


The Properties window changes automatically to show the properties available for the new TrendChart object.

16. In the Properties window, scroll down and click the value next to the commobj property. Make sure the communication object shown is the one you want to use.

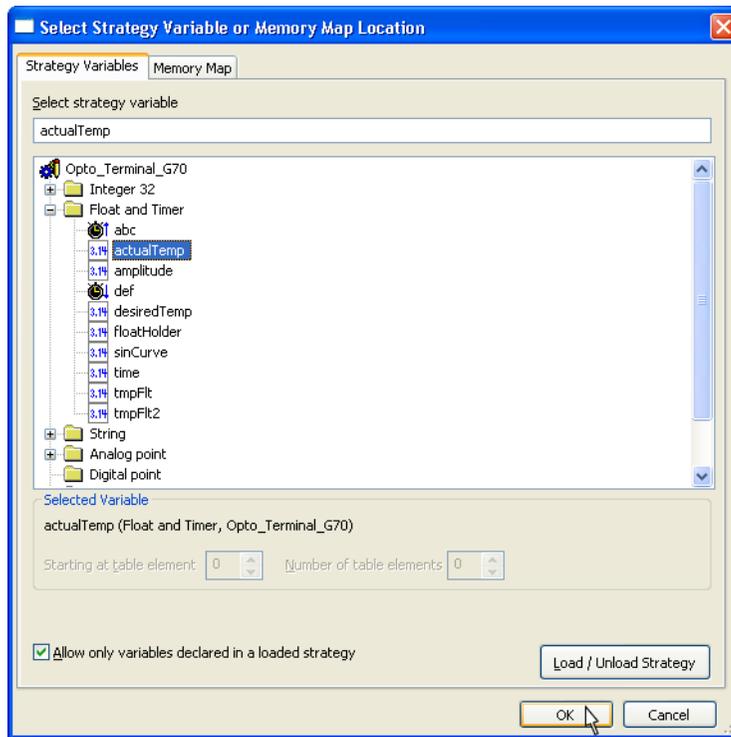
If you have added more than one controller or I/O unit, you can choose the one to use with this object from the drop-down list.

17. In the Properties window, scroll down and maximize the *Main* section. Click the value next to the tag property. Click the browse button (the small gray box):



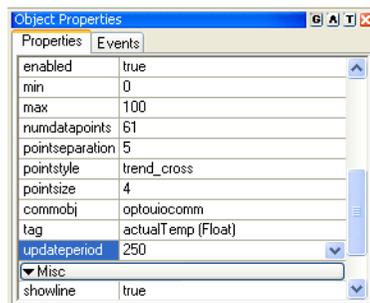
The PAC Control Variable dialog box opens. Since Qlarity Foundry applications can communicate with multiple PAC Control strategies as well as multiple controllers or I/O units, this dialog box lets you choose the strategy, the data type, and the specific variable within the strategy that this object (the TrendChart) should be associated with.

18. From the Select Variable or Memory Location window, select the float variable actualTemp.



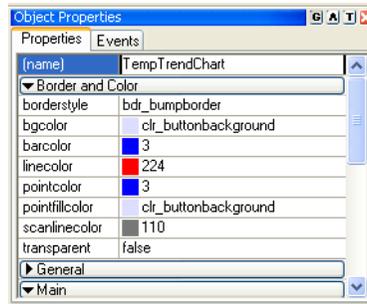
19. Click OK.

20. With the TrendChart object still selected, click the value next to the property updateperiod in the Properties window and enter a new value of 250.

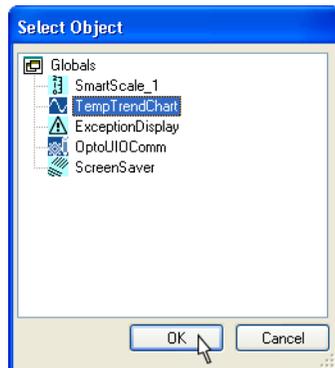


The update period determines how often, in milliseconds, the data for this object is read from or written to the strategy. An update period of 250 ms, or four times every second, is a fairly fast time for accessing strategy data.

21. With the TempTrendChart object still selected, change the (name) property to TempTrendChart.



22. Now click the SmartScaleV2 object  in the Object palette. Move the cursor to the simulated display area, and click and drag the mouse to the left of the TrendChart object to draw a small rectangle.
23. Select the SmartScale in the strategy tree. In the Properties window next to the property buddy click the browse button and select TempTrendChart, as shown:



24. Select File → Save to save your work again.
25. Keep the Qlarity Foundry application open in Microsoft Windows.

## Configuring the OptoTerminal for Downloading

*NOTE: To complete this section, you must have assigned an IP address to the OptoTerminal as described in “Configuring an OptoTerminal” on page 4. If you have already set the OptoTerminal to Download mode, skip this section and go on to the next section, “Downloading and Running the Application.”*

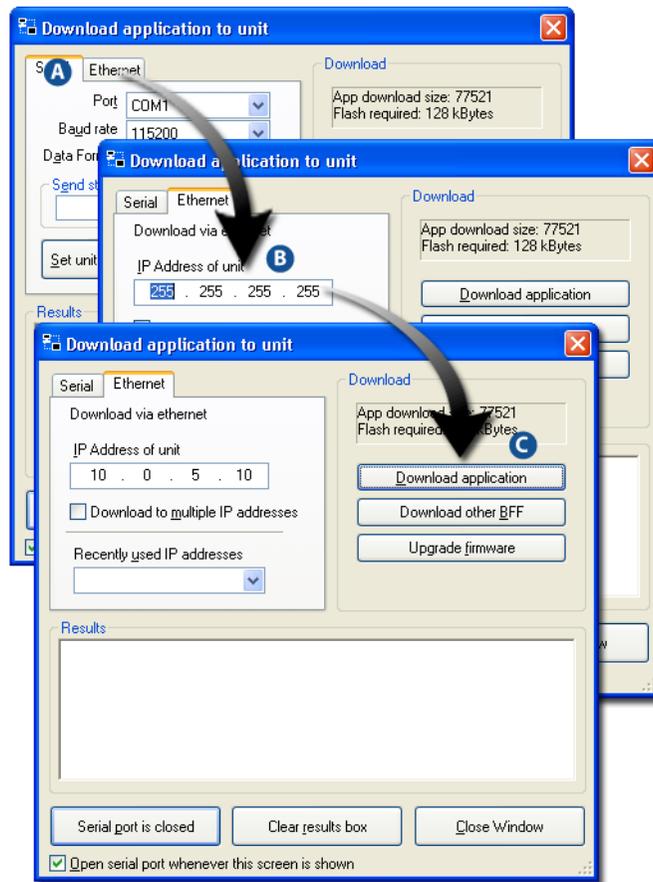
1. Make sure that the OptoTerminal is turned off.
2. Press and hold a finger or thumb on the upper left-hand corner of the screen.



3. While holding your finger on the screen, connect power to the terminal. Release your finger after the Power On Setup screen appears.
4. Press the Back or Next buttons at the bottom of the screen until FLASH MEMORY is highlighted, and then press Select.  
App Mode is highlighted.
5. Press Select. Then press Next until Download appears.
6. Press Select again.  
App Mode is highlighted.
7. Press Back or Next until DONE is highlighted, and then press Select.  
Save and Exit is highlighted.
8. Press Select to save your configuration changes.  
The terminal is reset. You can now download your application from Qlarity Foundry to the terminal.

## Downloading and Running the Application

1. In Qlarity Foundry, choose File→Download Application.  
After a few moments, the dialog box Download Application to Unit appears.
2. In this dialog box, do the following:
  - a. Click the Ethernet tab to view IP address settings.
  - b. In the IP Address field, enter the IP address of your OptoTerminal.
  - c. Click Download Application to send the application to the OptoTerminal.



3. Click Close Window after the application has been downloaded to the OptoTerminal. After a few moments, the Qlarity Foundry application starts running on the OptoTerminal.

## What's Next?

This quick start and sample applications are only a simple beginning; you can build very complex, powerful applications with Qlarity Foundry and PAC Control.

**For additional information on PAC Control**, see the *PAC Control User's Guide* (form 1700) and the *PAC Control Command Reference* (form 1701). These guides are on the CD that came with your SNAP PAC controller.

**To learn more about programming with Qlarity Foundry**, see the *OptoTerminal Qlarity Foundry User's Manual* (form 1344) and the *OptoTerminal Programmer's Reference* (form 1345). These manuals were shipped as printed documents with your OptoTerminal-G70 or -G75.

All guides are also available for download from our Web site, [www.opto22.com](http://www.opto22.com). The easiest way to find one is to search on its form number.

## Quick Start for I/O Units

This section steps you through using Qlarity Foundry software to create an OptoTerminal application for use with a SNAP PAC R-series I/O unit that is *not* running a PAC Control strategy. Used in this way, the OptoTerminal reads and writes data to the I/O unit's memory map. In this quick start, you'll do the following:

- Create a new Qlarity Foundry workspace and add objects from the Opto 22 object library.
- Test your application in Simulation mode.
- Link objects by building events.
- Download your Qlarity Foundry application to the OptoTerminal.

### Hardware and Software Requirements

To complete this quick start, you need the following:

- **SNAP PAC R-series controller and SNAP rack**
  - Mount the SNAP PAC controller on the rack and connect it to your computer, following instructions in Opto 22 form 1595, the *SNAP PAC R-Series User's Guide*.
  - Assign an IP address to the SNAP PAC controller following the instructions in Opto 22 form 1704, the *PAC Manager User's Guide*.

*NOTE: For this quick start, we'll assume that specific SNAP I/O modules are installed on the SNAP rack. You can still use the quick start even if these modules are not installed, however.*

- **OptoTerminal-G70 or OptoTerminal-G75**—If you haven't already set up your OptoTerminal and connected it to an Ethernet network, see ["Configuring an OptoTerminal" on page 4](#) for instructions.

*NOTE: If you don't have an OptoTerminal available, you can still complete most of this quick start using the Simulation mode in Qlarity Foundry software.*

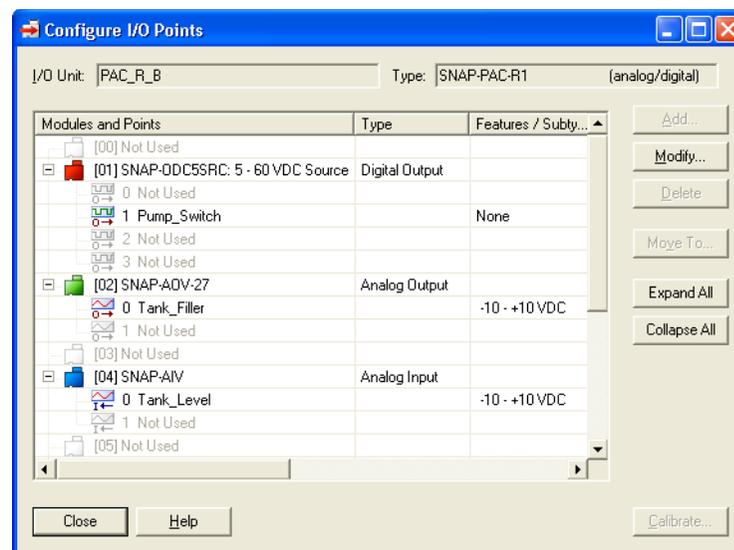
- **Qlarity Foundry software version 2.50 or newer**—If you haven't already installed Qlarity Foundry software on your computer, see ["Installing Qlarity Foundry Software" on page 6](#) for instructions.
- **Opto 22 PAC Manager software version 8.0 or newer**—PAC Manager software is on the CD that came with your SNAP PAC R-series controller. You can also download the latest version of PAC Manager from the Opto 22 Web site at [www.opto22.com](http://www.opto22.com).

### Preparing the I/O Unit

If you have a SNAP PAC R-series controller with the modules specified in the following steps, you'll be able to test your Qlarity application with real I/O. (If you have a SNAP PAC Learning Center, which includes these modules and a load panel, you can test it even more.) If you have a SNAP PAC

controller but do not have these modules, you can still follow the quick start and test your application using Simulation mode.

1. If PAC Manager is not already open, choose Start→Programs→Opto 22 and launch PAC Manager from a submenu.
2. Follow steps in the *PAC Manager User's Guide* (form #1704) to do the following:
  - Configure the I/O unit for the SNAP PAC R-series model you have.
  - Configure the module in position 01 (not zero) as a SNAP-ODC5SRC digital output module. Configure point 01 on this module as a digital output point with the name Pump\_Switch.
  - Configure the module in position 02 as a SNAP-AOV-27 analog output module. Configure point 00 on this module as an analog output point named Tank\_Filler.
  - Configure the module in position 04 as a SNAP-AIV analog input module. Configure point 00 on this module as a scalable analog input point named Tank\_Level, and give it custom scaling of 0 to 10 (rather than the default -10 to +10).
  - When you have finished, your PAC Manager configuration window should look like this:

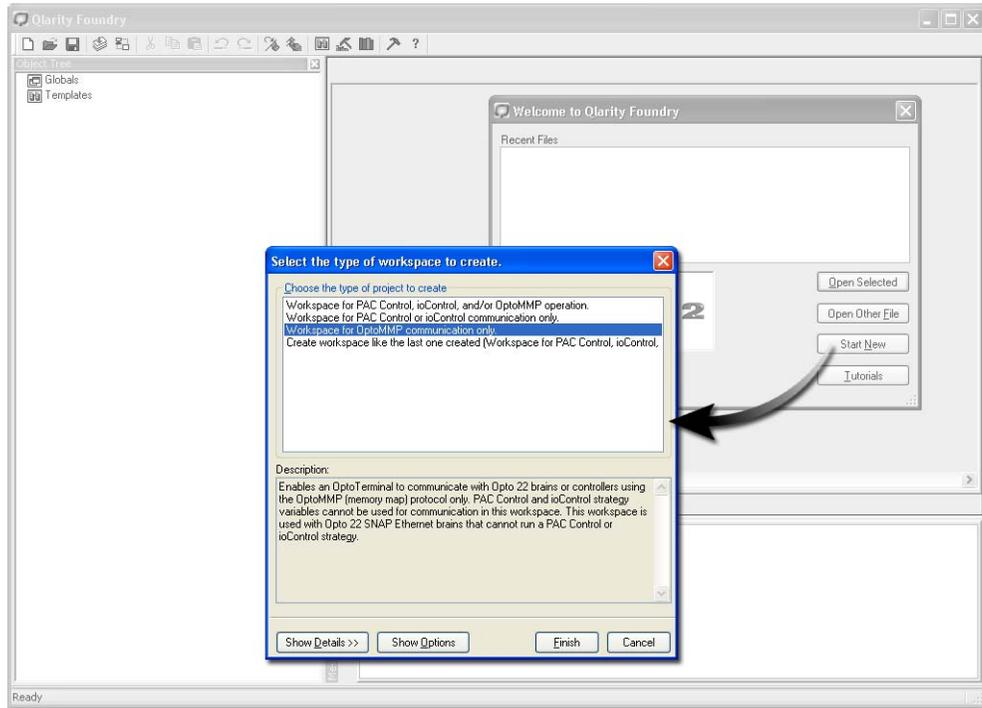


- Click Close.
  - Choose Tools→Send Configuration to I/O Unit and be sure to check Save to Flash. Send the configuration to the I/O unit.
3. Leave PAC Manager open.

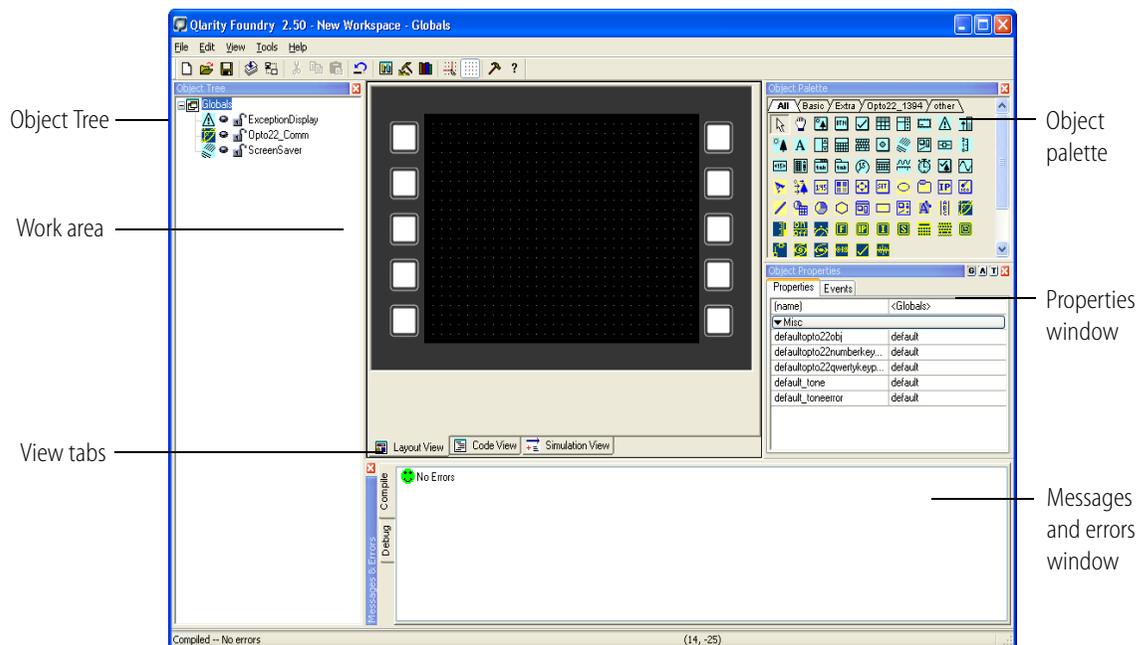
## Creating the Qlarity Foundry Application

1. From the Windows Start menu, choose Programs→Opto 22→Qlarity Foundry→Qlarity Foundry.

The Qlarity Foundry main window opens. If the Select Workspace dialog box is not open, click Start New to open it:



2. In the Select Workspace dialog box, choose Workspace for OptoMMP communication only.
3. Click Next and notice the options available, but don't change any of them now.  
For more information on these options, see the *OptoTerminal Qlarity Foundry User's Manual*.
4. Click Finish.  
The Qlarity workspace opens.



5. Notice the main elements in the workspace:

**Object Tree**—Any object you create in Qlarity Foundry is listed here. Some objects may be automatically added to a workspace when it is created, including the following:

- ExceptionDisplay—displays unhandled Qlarity exceptions (errors)
- Opto22\_Comm—handles the communication between the computer and/or terminal and the Opto 22 I/O unit.

**Work area**—This area looks like the OptoTerminal display. You place objects here as you create your application.

**View tabs**—Click a tab to change the view of the work area.

- Layout View (shown) is a graphics mode used to lay out the display. Generally, unless you are making changes to the programming code (or running your project in simulation mode), you will work in Layout View.
- Code View is used to write or edit an object’s programming code. Typically you use this mode only if you are customizing objects or creating your own objects.
- Simulation mode lets you connect to your I/O unit and test your display without needing to have an OptoTerminal.

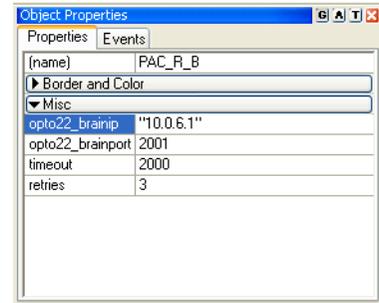
**Object palette**—Shows icons for all available objects. To see an icon’s name and description, move the mouse over it. To add an object, first click its icon, then draw in the work area. Click a tab to see icons in an individual library, or click the All tab to see all available icons.

**Properties window**—When you click an object in the Object Tree or in the work area, its properties are listed here. You can change a property by clicking on its value and typing in a new value or choosing a value from a drop-down list. To see a property’s description, move the mouse over its name.

**Messages and errors window**—Displays errors or other messages generated when the workspace is compiled or during a debug operation. These messages can help you identify and correct problems with the workspace.

6. In the Object Tree, click the Opto22\_Comm object to set communication properties. Notice that the list in the Properties window changes.
7. In the Properties window, click the value next to the (name) property, and type in the name of the I/O unit. (In this example, we've used PAC\_R\_B as the name.)
8. Click the value next to the property opto22\_brainip and then enter the IP address of the SNAP PAC R I/O unit you're using for this exercise, as shown at right.

You do not need to enter the quotation marks; they are entered for you.



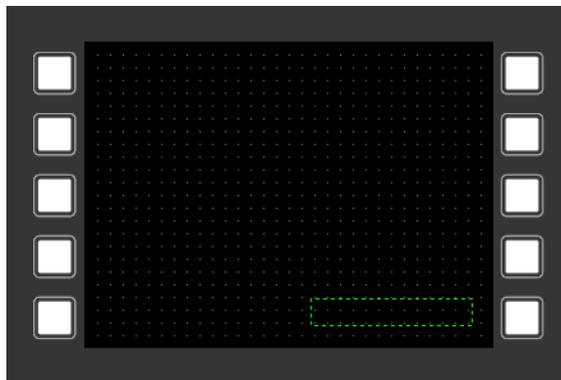
*NOTE: You can add additional Opto22\_Comm objects if your OptoTerminal needs to communicate with additional I/O units.*

9. Choose File→Save Workspace to save your work to a project file. In the Save As dialog box, give the file a name, navigate to where you want it saved, and then click Save.

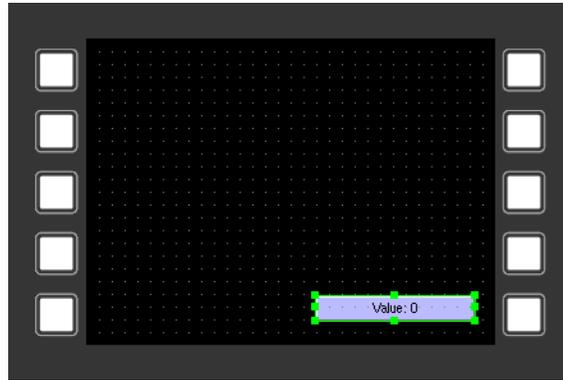
### Displaying a Value from the Memory Map

A wide assortment of data is available in the I/O unit's memory map, and you can use the OptoTerminal to display that data as necessary. As a simple example, let's display the digital scan counter, which just counts the number of times the SNAP brain scans the digital points connected to it on the mounting rack. Since the brain assumes all module positions contain digital modules unless otherwise configured, you don't need to have an actual module present to see the scan counter at work.

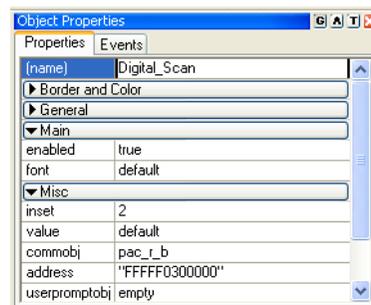
1. In the Object palette, click the OptoMemMapInteger object . Move your cursor to the work area and, holding down the left mouse button, draw a rectangle near the lower-right corner, as shown below:



2. When you release the mouse button, a solid rectangle appears on the screen with a green line around it.



3. Notice that the Properties window changes; it now shows properties for the new object. Click the (name) property and type `Digital_Scan` as the name for this object. (Make sure you type the underscore, or the name is rejected.)

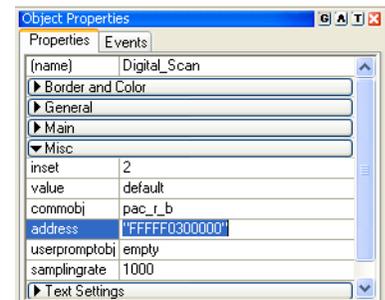


4. Next, scroll down to the caption property and type `Digital Scan:` as the value.
5. Scroll down again; check the `commobj` property to make sure it shows the correct I/O unit. If you have more than one communication object, you can choose the correct one for this object from the drop-down list.
6. In the address property, highlight the address value.

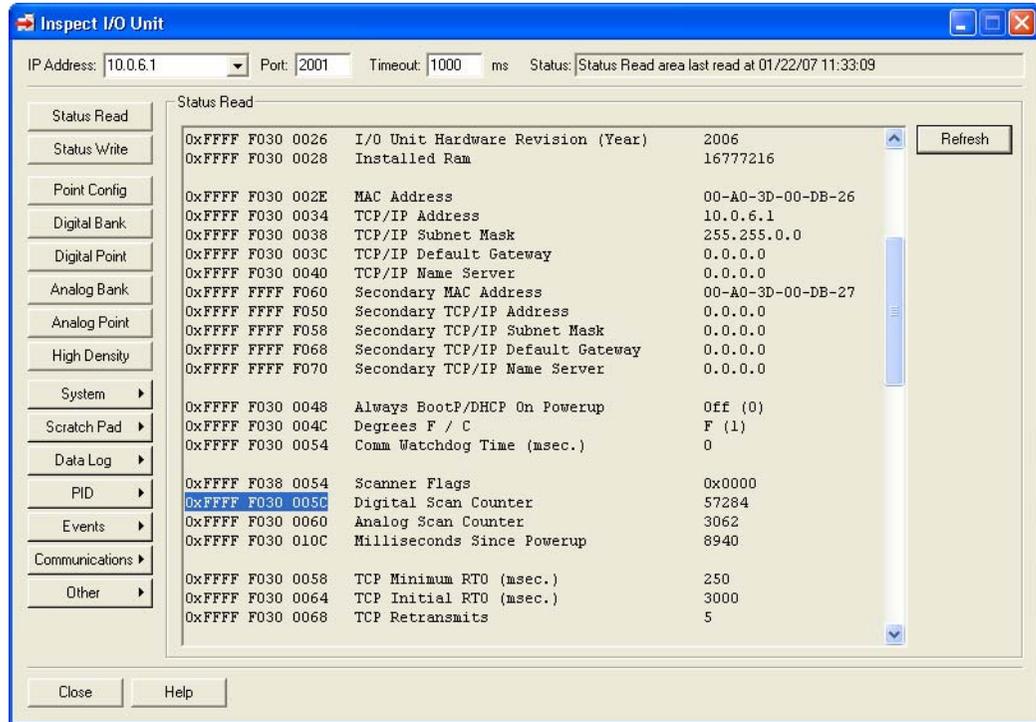
There are two ways to find out a memory map address:

- Look in the memory map appendix of Opto 22 form #1465, the *OptoMMP Protocol Guide*.
- Inspect the I/O unit in PAC Manager and copy the address from there.

Since PAC Manager is already open, we'll use the second way.

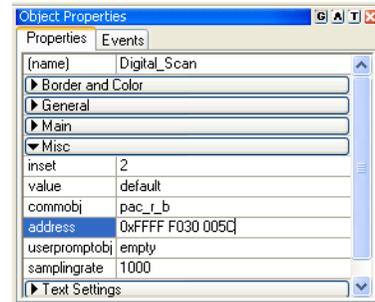


7. In PAC Manager, click the Inspect icon . In the IP Address field, type the IP address of the I/O unit (or choose it from the drop-down list). Click Status Read.
8. In the Status Read area, scroll down until you see Digital Scan Counter. Use your mouse to highlight the entire address in the left-hand column.

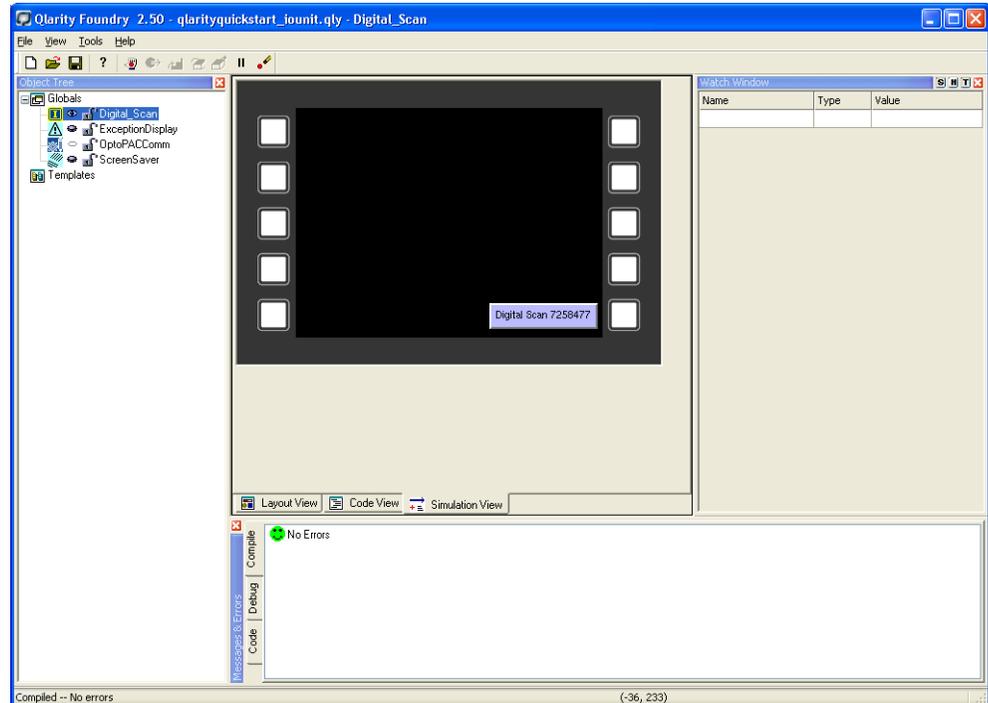


- Press CTRL-C to copy the address.
- Back in Qlarity Foundry, press CTRL-V to paste this memory map address in the address field, as shown at right.

Notice that you can enter the address as copied, 0xFFFF F030 005C, or you can enter it without the hex indicator and without the spaces, like this: FFFF030005C.



- Choose File→Save Workspace.  
Now let's check this object in Simulation mode to see how it will work.
- Click the Simulation tab at the bottom of the work area to change the view.  
You see the work area as it will look on the OptoTerminal display:



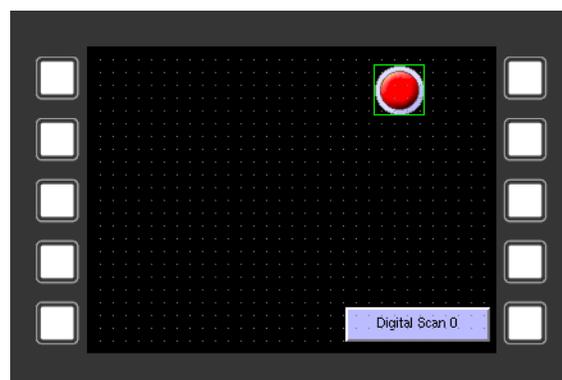
If your I/O unit is active on the network, you'll see the digital scan counter value increasing as the brain scans module positions on the rack.

Now, let's add a button to the display.

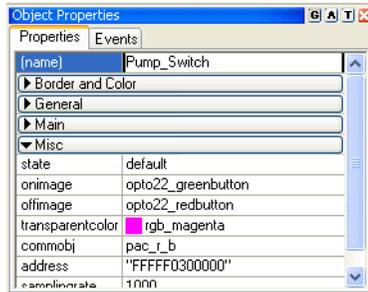
### Adding a Button

One of the most common needs in an operator interface is a button for the operator to push. Here's how to add a button that toggles between two states (on and off, open and closed, etc.).

1. Click the Layout View tab to return to work.
2. In the Object palette, click the Opto22\_ToggleButton . Move your cursor to the work area and draw a small square near the upper-right corner. When you release the mouse button, a button appears on the screen.

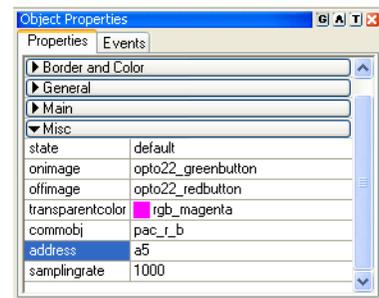


- In the Properties window, click the (name) property and type `Pump_Switch` as the name for this button. (Remember the space will be replaced with an underscore.)



- Scroll down to the address property and highlight the address value.

If the data you are reading or writing to in the memory map is the status of a digital point or the value of an analog point, you don't have to use the complete memory map address. Instead, you can type the point number (for example, `a5` for digital point five or `a16` for analog point 16).



For this quick start, enter `a5` as shown at right.

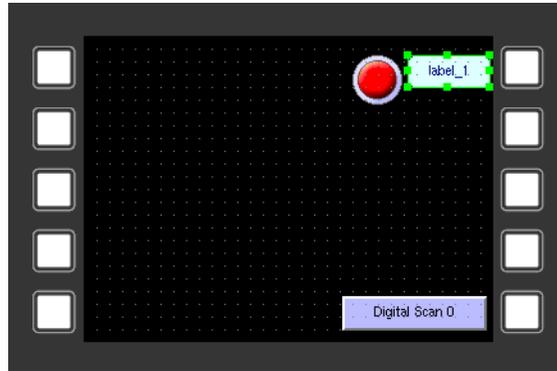
- Choose File → Save Workspace.  
The button you've created will turn the digital point 01 on the module in position 01 on or off. You can test it with your SNAP PAC R.
- Click the Simulation tab.
- Use the mouse cursor to push the button.

The button changes to green on the screen, indicating that the digital point it controls is now on. If you have a SNAP-ODC5SRC module in position 01 on the I/O unit, you'll see that the module's LED for the point is lit. If you have a SNAP Learning Center, you'll see Inside Light LED on the front panel is also lit.

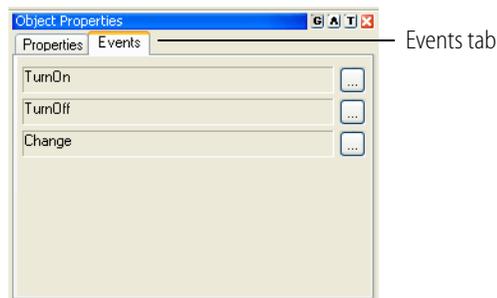
### Changing the Display Based on the Button

In the last section, you created a button on the OptoTerminal display that changed a digital output point on the I/O unit. Now let's also make it change another object on the display. To do so, we have to create an *event* in the Qlarity application.

- Click the Layout View tab.
- In the Object palette under the Basic tab, click the LabelV2 icon . In the work area, draw a small rectangle at the right side of the button.



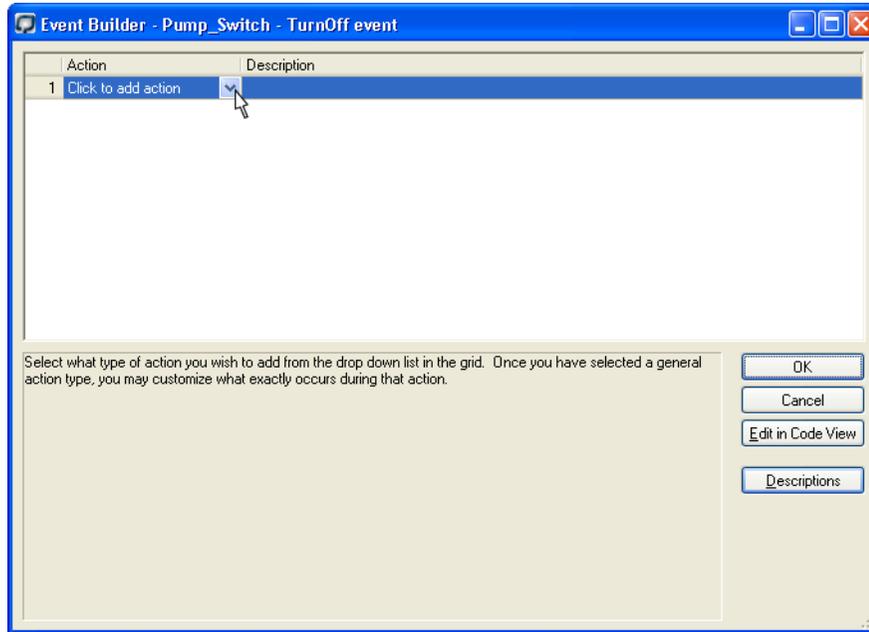
3. In the Properties window for the label, change the following properties:
  - (name): `Pump_Status`
  - bgcolor: `rgb_red`
  - value: `Pump Off`
4. In the work area, click back on the Pump\_Switch button. In its Properties window, click the Events tab.



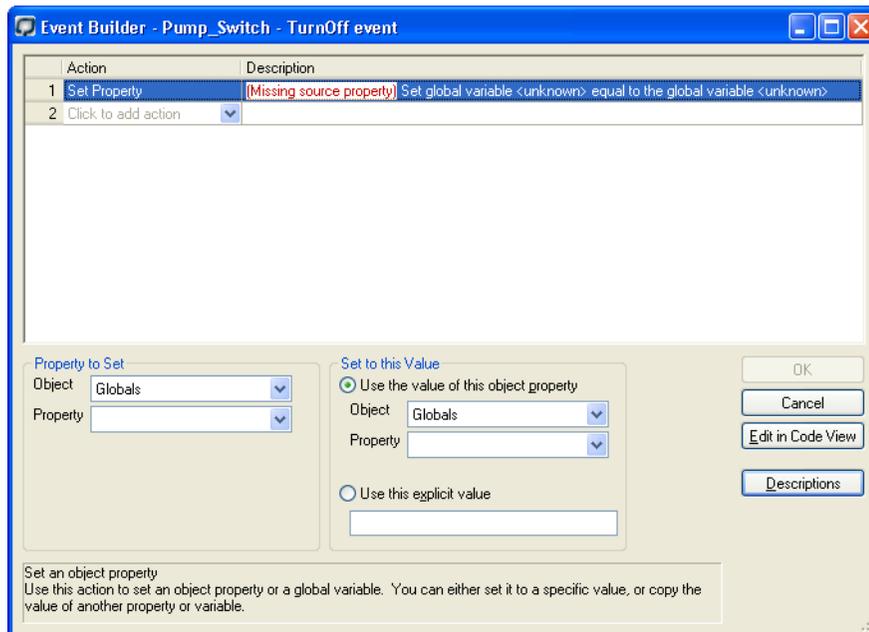
5. Click the browse button  next to TurnOff.

The Event Builder window opens. You use the Event Builder to create logic within your Qlarity application, so that what appears on the display is in a form that the operator who uses it can understand.

For example, this button changes the state of digital point 01 on the module in position 01, but the operator who pushes it shouldn't have to remember what the button does or what point 01 on module 01 controls. It is better to provide a text label for the button, so the operator can see what the button will do.



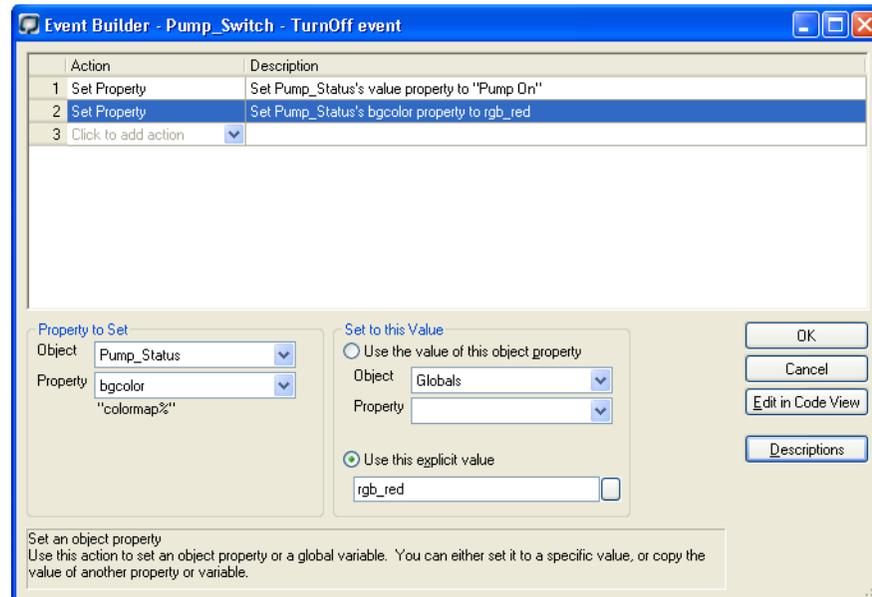
6. Click the down arrow next to Click to add action. In the drop-down menu, choose Set Property.



7. Do the following:
  - In the Property to Set area, choose Pump\_Status as the Object and value as the Property.
  - In the Set to this Value area, choose Use this explicit value and enter Pump Off in the field.
8. For the second action, click the down arrow next to Click to add action and choose Set Property again. Do the following:

- In the Property to Set area, choose `Pump_Status` as the Object and `bgcolor` as the Property.
- In the Set to this Value area, choose Use this explicit value and choose `rgb_red` from the pop-up list.

The Event Builder window now looks like this:



9. Click OK.  
You have built an event: when the `Pump_Switch` button is toggled off, the color and text of `Pump_Status` will change. Now you also need to build an event that states what will happen when the `Pump_Switch` button is toggled on, or else it will always remain red.
10. In the Properties window, with the Events tab open, click the browse button next to `TurnOn` to open the Event Builder again.
  - Click the down arrow next to `Click to add action` and choose `Set Property`. In the Property to Set area, choose `Pump_Status` as the Object and `caption` as the Property. In the Set to this Value area, enter the explicit value `PUMP ON`.
  - Click the down arrow next to `Click to add action` again and choose `Set Property`. In the Property to Set area, choose `Pump_Status` as the Object and `bgcolor` as the Property. In the Set to this Value area, choose the explicit value `rgb_green`.
  - Click OK.
11. Save the workspace.
12. Click the Simulation tab to test your changes.

You now see both the button and the label. When you push the button, the label changes from red to green and the text on it changes as well. Pushing the button now changes *both* the digital output point 5 and the OptoTerminal display.

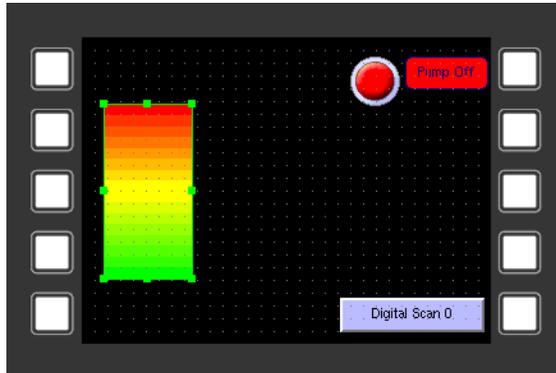
*NOTE: In this sample, the pump status is tied to the button, not to an actual pump. To monitor the status of the pump itself, do not use the generic LabelV2 object; instead, use an OptoTextToggleButton object and set its address property to indicate the memory map address for the state of the pump.*

### Using a Slider to Change the Display

Now we'll build another object for an operator to interact with on the OptoTerminal display.

*NOTE: This object will not work unless your I/O unit has the analog output module you configured installed in the correct position. However, you may still want to follow the steps to see how the objects are created and linked.*

1. Click the Layout View tab. In the Object palette, choose the Opto22\_Gauge  and draw a vertical rectangle on the left side of the work area. Leave some space at the bottom.

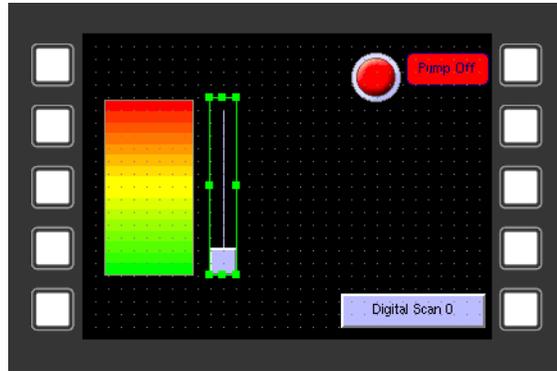


2. In the Properties window for the gauge, change the following properties:
  - (name): Tank\_Level
  - endrange: 10
  - address: a16
  - samplingrate: 0

We set the endrange to 10 (and left the starange at 0) to match the custom scaling set up for point a16 when the analog output module was configured on [page 21](#).

By setting the sampling rate to zero, we've chosen not to use the value from the analog point as the input for this gauge. Instead, the operator will input the value using a slider.

3. In the Object palette, choose the OptoSlider  and draw another rectangle, the same length but narrower, next to the gauge.



4. In the Properties window for the slider, change the following properties:
  - (name): `Tank_Filler`
  - endrange: 10
  - address: a8
5. In the Properties window, click the Events tab and then click the browse button next to Change.
6. In the Event Builder, do the following:
  - Add the action Set Property.
  - In the Property to Set area, choose `Tank_Level` as the Object and `value` as the Property.
  - In the Set to this Value area, choose Use the value of the object property. Choose `Tank_Filler` as the Object and `value` as the Property.
  - Click OK.
7. Save your workspace. If your I/O unit has the module installed in the correct position as configured, click the Simulation tab to test the slider.

When you use the mouse to move the slider up and down, the gauge fills and empties.

*NOTE: If your I/O unit does not have the module installed, the slider's handle will be missing and the slider won't work.*

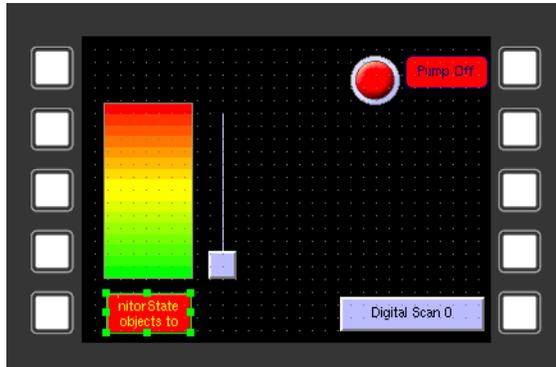
### Working with a Container Object

Now let's build a more complex indicator that identifies what the gauge is monitoring (a tank) and changes based on the tank's status. You'll create this indicator by using an object that contains other objects.

*NOTE: Testing this object also requires the configured module to be installed on your I/O unit.*

1. Click the Layout View tab.

2. In the Object palette, click the OptoStateMonitor container  and draw a small rectangle below the gauge:



3. In the Properties window for the container, change the following properties:
  - (name): `Monitor_Fill`
  - address: `a8`

Next, we'll add three objects within the container. We'll set the properties of these objects so that the text and colors change based upon the value of analog output point 08, which Tank\_Filler (the slider) controls. This portion of the display describes in words what the gauge represents graphically.

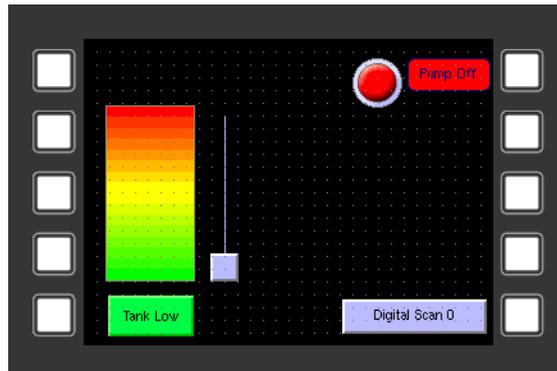
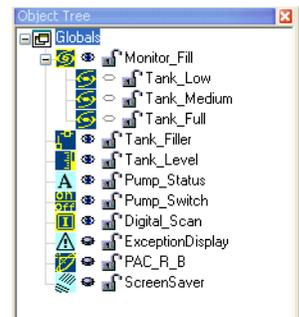
4. In the Object palette, click the OptoStateMonitorState object  and draw a rectangle anywhere inside the Monitor\_Fill container. Change its properties as follows:
  - (name): `Tank_Low`
  - caption: `Tank low`
  - rangemax: `4`
  - bgcolor: `rgb_green`
5. Again using the OptoStateMonitorState object , draw another rectangle inside the Monitor\_Fill container. Change its properties as follows:
  - (name): `Tank_Medium`
  - caption: `Tank OK`
  - rangemax: `8`
  - bgcolor: `rgb_yellow`
6. Draw a third OptoStateMonitorState object inside the Monitor\_Fill container. Change its properties as follows:
  - (name): `Tank_Full`
  - caption: `Tank full`
  - rangemax: `10`

– bgcolor: rgb\_red

7. Notice that Tank\_Low, Tank\_Medium, and Tank\_Full appear in the Object tree under Monitor\_Fill, as shown at right, because they are contained within Monitor\_Fill.

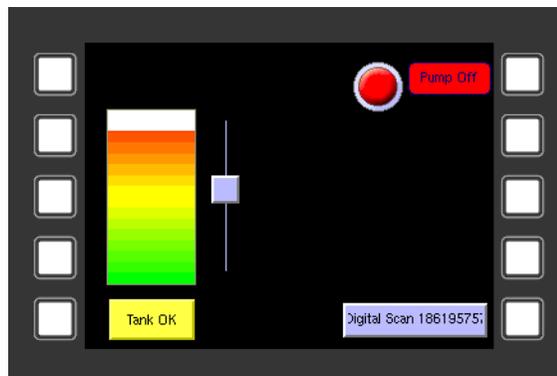
If you need to change properties for any of these objects (or for the container), click on the object's name in the Object Tree and it will appear in the Properties window.

Your Qlarity Foundry application should now look something like this:



8. Save your workspace. If your I/O unit has the module installed in the correct position as configured, click the Simulation tab to test the display.

When you move the slider up, the gauge fills, and the label changes color and text to indicate the tank level.



If you are testing on a SNAP PAC Learning Center, the meter on the load panel also changes to indicate the tank level.

*NOTE: If your I/O unit does not have the module installed, the slider's handle will be missing and the label won't change.*

Now that your Qlarity Foundry application is created and tested in Simulation mode, it's time to download it to the OptoTerminal.

9. Keep Qlarity Foundry open.

## Configuring the OptoTerminal for Downloading

*NOTE: To complete this section, you must have assigned an IP address to the OptoTerminal as described in "Configuring an OptoTerminal" on page 4. If you have already set the OptoTerminal to Download mode, skip this section and go on to the next section, "Downloading and Running the Application."*

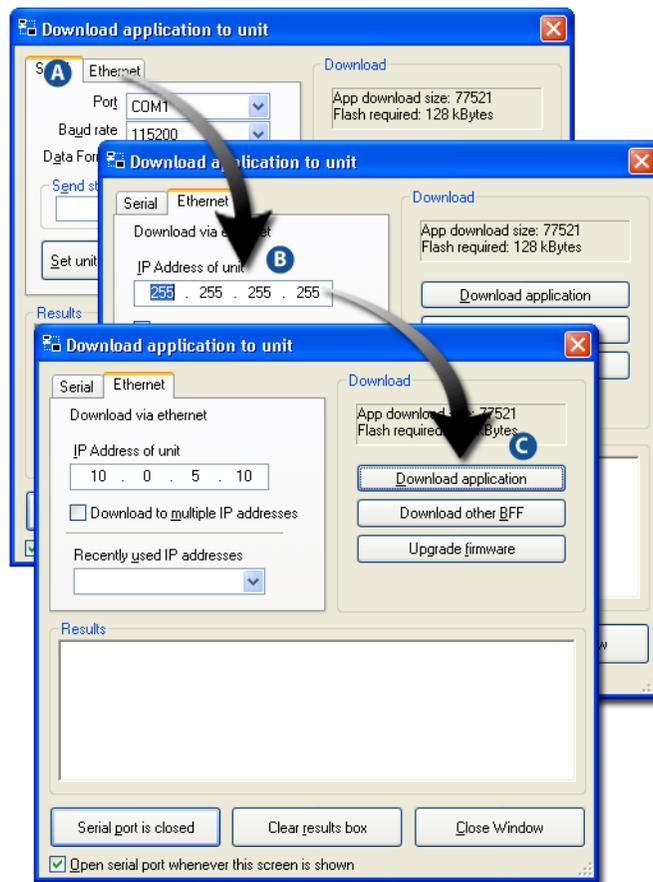
1. Make sure that the OptoTerminal is turned off.
2. Press and hold a finger or thumb on the upper left-hand corner of the screen.



3. While holding your finger on the screen, connect power to the terminal. Release your finger after the Power On Setup screen appears.
4. Press the Back or Next buttons at the bottom of the screen until FLASH MEMORY is highlighted, and then press Select.  
App Mode is highlighted.
5. Press Select. Then press Next until Download appears.
6. Press Select again.  
App Mode is highlighted.
7. Press Back or Next until DONE is highlighted, and then press Select.  
Save and Exit is highlighted.
8. Press Select to save your configuration changes.  
The terminal is reset. You can now download your application from Qlarity Foundry to the terminal.

## Downloading and Running the Application

1. In Qlarity Foundry, choose File→Download Application.  
After a few moments, the dialog box Download Application to Unit appears.
2. In this dialog box, do the following:
  - a. Click the Ethernet tab to view IP address settings.
  - b. In the IP Address field, enter the IP address of your OptoTerminal.
  - c. Click Download Application to send the application to the OptoTerminal.



3. Click Close Window after the application has been downloaded to the OptoTerminal.  
After a few moments, the Qlarity Foundry application starts running on the OptoTerminal.

*NOTE: Again, the Pump On/Off button and the digital scan counter will work, with or without a module, but the Tank Filler and Tank Level indicators will not work unless you have the correct module installed on the I/O unit.*

## What's Next?

This quick start is only a simple beginning; you can build very complex, powerful applications with Qlarity Foundry.

To learn more about programming with Qlarity Foundry, see the *OptoTerminal Qlarity Foundry User's Manual* (form 1344) and the *OptoTerminal Programmer's Reference* (form 1345). These printed manuals were shipped with your OptoTerminal-G70 or -G75.

All guides are also available for download from our Web site, [www.opto22.com](http://www.opto22.com). The easiest way to find one is to search on its form number.

# Using Opto 22 Library Objects

## Introduction

Qlarity Foundry is a software development environment for building OptoTerminal *applications*, which are collections of code and graphics that are downloaded to and run on the terminal. This guide introduces only the basic use of Qlarity Foundry with Opto 22 hardware; for detailed information on using Qlarity Foundry, see the *OptoTerminal Qlarity Foundry User's Manual* and the *OptoTerminal Programmer's Reference Manual* in the included product documentation.

In order to simplify communications with Opto 22 devices, special communications and interface objects are provided. In Qlarity Foundry, *communications objects* are like the software drivers a computer uses to communicate with peripheral devices such as printers or modems. Interface objects are like the software applications that use the driver.

## Opto 22 Communications Objects

Each Opto 22 object library includes a communications object that defines which SNAP Ethernet-based hardware the OptoTerminal will communicate with. The Opto 22 PAC Object Library uses the OptoPACComm communications object, while the Memory Map (MMP) Object Library uses the Opto22\_Comm object.

Each communications object must be configured with network communications settings, including the IP address of the SNAP hardware (controller or I/O unit) that the OptoTerminal will communicate with.

One Opto 22 communications object is automatically added to each new workspace. If the OptoTerminal will communicate with more than one Opto 22 system, you must add an Opto 22 communications object for each system. A maximum of 64 communication objects (up to 32 of which can be PAC Control communication objects) can be added to a workspace, although using this many communication objects may make the terminal operate slowly.

## Opto 22 Object Libraries

As you have seen in the previous chapter, two interface object libraries are available for working with Opto 22 SNAP Ethernet-based hardware:

- The PAC Control Object Library is a collection of objects used to send and receive data from the variables (referred to as *tags* in Qlarity Foundry) in a PAC Control strategy running on a SNAP PAC controller.
- Like the PAC Control Object Library, the MemoryMap Object Library is a collection of objects used to send and receive data, but this object library works with data in the memory map of a SNAP PAC controller or I/O unit. The device's memory map stores device data, I/O point data for an I/O unit, and Scratch Pad data, which can be used for event/reactions on an I/O unit or for peer-to-peer communication among controllers.

### Choosing a Library

When you start Qlarity Foundry, you choose a workspace based on the data you want to read or write: either PAC Control strategy data or Memory Map data. The workspace automatically adds a communication object of the type you need and contains the object library for the data you want. Opto 22 object libraries appear in the Object palette as blue and yellow or blue and white icons.

### If You Need Both Libraries

If you need to read and write data to both PAC Control strategies and I/O unit memory maps in the same Qlarity Foundry application, you can do so. After choosing a workspace, simply add the other library by clicking the Libraries button  on the toolbar, then clicking the *Add existing library* button and choosing the additional library from the list. For more information on adding libraries, see the *OptoTerminal Qlarity Foundry User's Manual*.

Once both libraries are in your workspace, you can choose communication objects and interface objects from both as needed.

## Common Uses for Opto 22 Library Objects

These instructions presume that you have already added to your Qlarity Foundry project either the Opto22Obj communications object or the OptoPACComm communications object. If your OptoTerminal application needs to communicate with more than one SNAP controller, SNAP I/O unit, or SNAP-IT unit, you must add additional communications objects to your project.

To do this...	Use the Opto 22 object(s):		Notes
	PAC Control objects	Memory map objects	
Turn a digital I/O point on and off	 PACToggleButtonV2  PACToggleBitmapV2	 Opto22_ToggleButton  OptoTextToggleButton	Choose one depending on how you want the value to appear on screen.
Display the value of an analog input point.	 PACTachometerV2  PACGaugeV2	 Opto22_Tachometer  Opto22_Gauge	Choose one depending on how you want the value to appear on screen.
Change an analog value	 PACSliderV2	 OptoSlider	
Trigger an alarm based on the state of a digital point or analog value	 PACStateAlarmV2	 OptoStateMonitor	
Display the state of a digital point	 PACToggleButtonV2	 Opto22_ToggleButton	
Change a graphic based on the state of a digital point.	 PACToggleBitmapV2	 Opto22_TextToggleButton	
Display the value of a variable in a PAC Control strategy or a memory map location	 PACLabelV2  PACFastLabelV2	 OptoMemoryMapFloat  OptoMemoryMapString  OptoMemoryMapInteger	Choose one depending on the type of data and the frequency you want it displayed.

## Opto 22 PAC Control Library Object Reference

Opto 22 PAC Control library objects are listed below. Refer to the online help in Qlarity Foundry for parameters and detailed information about each object.

*NOTE: Some objects allow the user to define the displayed text font. In this case, you can choose either a BDF or TTF font. The BDF (Adobe Glyph Bitmap Distribution Format) font uses a raster, or bitmapped, font to display the text. The TTF font uses a TrueType font to display the text. Choose the BDF option unless you want to add a specific font as a resource for your Qlarity Foundry project.*

Opto 22 PAC Control Library Objects	
 <p><b>OptoPACComm</b> Coordinates communication between an OptoTerminal and an Opto 22 SNAP PAC controller or I/O unit running a PAC Control strategy. To communicate with an Opto 22 device, configure this object's properties (such as the controller's IP Address). Then create other PAC objects, such as the PACTrendchartV2. Set up the commObj and tag properties of these objects in order to read or set a variable (tag) in your PAC Control workspace.</p>	 <p><b>PACDigitalGridV2</b> This object reads a 32-bit or 64-bit mask from an Opto 22 device. Each bit is displayed in the grid as an on/off value.</p>
 <p><b>PACDataLoggerV2</b> This object simplifies the process of logging data from the Opto device. The value determined by the tag property is logged to a file at a rate specified by the Sampling rate property. Alternatively, the appendline method can be called to append a line to the file. A call to readline will give the data stored on a given line number in the file. If the filename is invalid when a read or write occurs then an exception is thrown.</p>	 <p><b>PACGaugeV2</b> This object implements a flexible bar graph to graphically indicate the level of a floating point value. The object supports user-defined start- and end-points. Because this object displays a value, it queries a new value from the Opto tag whenever it becomes enabled or whenever commobj or tag properties are changed.</p>
 <p><b>PACLabelV2</b> Displays a value read from a tag from a PAC control strategy. The value read is displayed according to the prefix, value and suffix properties. The value property is interpreted based on the data type set in the tag property.</p>	 <p><b>PACFastLabelV2</b> This object is a quick reacting label object. If you are attempting to display rapidly changing data, this object is generally a better choice than the other label objects.</p>
 <p><b>PACListBoxV2</b> Displays any number of text strings and optionally allows the user to select one of them. If there are more strings than can be displayed at one time, scroll buttons allow the user to scroll the view. To change the value of any element in the listbox, the item must be selected and then pressed one additional time. An override function is called whenever the selected item in the PACListBoxV2 changes (either programmatically or due to the user touching the box).</p>	 <p><b>PACQwertyKeypadV2</b> This object is a keypad that allows the user to send data to a tag on an Opto 22 device when the enter button is pressed. The data-type of the tag property of this object determines how the data is sent to the device. If the data-type of the value entered into this object does not match the data-type of the tag property of this object, then nothing is sent to the Opto 22 device.</p>
 <p><b>PACSliderV2</b> Implements a slider control, which has a square position grip (the "button") and a linear range for setting the grip. When the current position is changed, the new value is sent to the Opto device. Data queried from the Opto device will update the position of the slider.</p>	 <p><b>PACPlotV2</b> This object implements a type of 2D graph that can be used to plot recent values on a X-Y axis. This object is robust and can utilize quite a bit of processing power. To compensate for this, you should minimize using other polling objects on the same screen as a plot object.</p>
 <p><b>PACSpinnerV2</b> A control object that lets the user increment the number displayed by pressing the up or down arrows. Each time the value is changed it is sent to the Opto device. Any data sent to this object from the Opto device will change the value of the spinner.</p>	 <p><b>PACStateAlarmV2</b> This object is used to monitor a value queried from the Opto 22 tag. Based on the current value of the tag, the StateAlarm can change captions, colors and optionally blink. To configure the various captions/colors for different value ranges, set the StateInformation property. Because this object displays a value, it queries a new value from the Opto 22 tag whenever it becomes enabled or whenever commobj or tag properties are changed.</p>
 <p><b>PACTachometerV2</b> Displays a value in the form of a dial gauge. Because this object displays a value, it queries a new value from the Opto 22 tag whenever it becomes enabled or whenever commobj or tag properties are changed.</p>	 <p><b>PACNumberKeypadV2</b> Displays a simple keypad that the user can use for numeric entry. The object is geared to be a pop-up object that can be "popped up" whenever numeric entry is necessary.</p>

Opto 22 PAC Control Library Objects	
 <p><b>PACToggleButtonV2</b> Represents a Boolean value. You can touch this object to toggle the checked value of this object on and off. An override is called whenever the state of this object changes. This object indicates the last value queried from the Opto 22 device, the state of the object from the last press, or the last value assigned to the value property. If the state is changed by a screen press or a value is assigned to the value property, then the new value is sent to the Opto 22 device. If the user-settable property is set to false, then this object is assumed to display values only and not set values, so it queries a new value from the Opto 22 device whenever it becomes enabled or whenever commobj or tag properties are changed.</p>	 <p><b>PACVirtualVariableV2</b> This object is a simple interface for non-Opto 22 objects to communicate with an Opto 22 device. The tag property must be set either at design-time or at run-time by a call to the CreateTag method in the PACComm object. This object will query the tag in the tag property by calling the queryValue method. When the response is received from the query, the appropriate QuerySucceeded method is called for the tag datatype. For example, if the data type is an _OPTOFLOAT, then the QueryFloatSucceeded method is called. If the query failed for some reason, the QueryFailed method is called. To write a value to the Opto 22 device, call the appropriate Write function according to the data type of the tag (for example, call the WriteFloat method if the data type is _OPTOFLOAT). If the write succeeds, then the WriteSucceeded method is called. Otherwise, the WriteFailed method is called. Each read or write is associated with an ID number. When the Write or Query methods are called, a number is returned that is used to identify the transaction. A number is passed into the appropriate QuerySucceeded or WriteSucceeded methods that indicates the ID number that the method is being called in response to.</p>
 <p><b>PACTrendChartV2</b> This object can be used to query and chart values from an Opto 22 device. These values are displayed based on the ChartType, PointSeparation, and ConnectPoints properties. This object can display trending charts, scanning charts and bar charts.</p>	 <p><b>PACToggleBitmapV2</b> An on/off toggle switch with changeable bitmaps.</p>

## Opto 22 Memory Map Library Object Reference

Opto 22 library objects are listed below. Refer to the online help in Qlarity Foundry for parameters and detailed information about each object.

Opto 22 Memory Map Library Objects	
 <p><b>Opto22Obj</b> Handles all communications with one Opto 22 I/O unit (includes SNAP-IT systems). Set the IP address of your Opto 22 device in this object, and then create other Opto 22 objects and point them to this one for communications.</p>	 <p><b>Opto22_Gauge</b> A flexible bar graph that indicates the level of a floating point value. The object supports user-defined start- and endpoints and an optional color gradient across the length of the bar. Tie this object to a read or read/write address on the Opto 22 device. If the location you tie this object to is read/write, however, you can only display the value; you cannot use the value property to set the location.</p>
 <p><b>Opto22_ToggleButton</b> An on/off toggle switch for digital points. Custom bitmaps can be used in place of the default buttons.</p>	 <p><b>Opto22_Tachometer</b> An analog dial gauge to display an analog value.</p>
 <p><b>OptoMemMapFloat</b> Sets or displays a floating point value. Configure this object with the appropriate memory map address for the floating point value. You can also set the associated Opto 22 memory map variable by setting the value property of this object. (Note: This property must be set at runtime; it cannot be initialized in Layout View.)</p>	 <p><b>OptoMemMapIPAddress</b> Sets or displays data in a memory map address that is in the format of an IP address. Most often used for display, but you can also set the associated memory map value by setting the value property of this object. (Note: This property must be set at runtime; it cannot be initialized in Layout View.) Do not use this object to change the IP address of an I/O unit.</p>

Opto 22 Memory Map Library Objects	
 <p><b>OptoMemMapInteger</b> Sets or displays an integer value. Configure this object with the appropriate memory map address for the integer value. You can also set the associated Opto 22 memory map variable by setting the value property of this object. (Note: This property must be set at runtime; it cannot be initialized in Layout View.)</p>	 <p><b>OptoMemMapString</b> Sets or displays a string value. Configure this object with the appropriate memory map address for the string value. You can also set the associated Opto 22 memory map variable by setting the value property of this object. (Note: This property must be set at runtime; it cannot be initialized in Layout View.)</p>
 <p><b>OptoNumberKeypad</b> Displays a simple keypad for entering numbers on screen. The object can work as a “pop-up” object that appears whenever numeric entry is necessary.</p>	 <p><b>OptoQwertyKeypad</b> Displays a simple, fixed-size keypad for entering text on screen. Enter and Cancel buttons can be used to trigger events that the program can use to perform actions.</p>
 <p><b>OptoRealTimeClock</b> Sets or displays the clock setting of an Opto 22 SNAP Ethernet-based I/O unit (includes SNAP-IT systems). Configure this object with the appropriate memory map address for the I/O unit’s real-time clock. You can also set the associated Opto 22 memory map variable by setting the value property of this object. (Note: This property must be set at runtime; it cannot be initialized in Layout View.)</p>	 <p><b>OptoSlider</b> Used to change an integer or float value. Drag the button on this object to change the value of the memory map location the object is tied to. You should tie this object only to a read/write location on the Opto 22 system. (Note: Analog outputs are considered read/write locations.)</p>
 <p><b>OptoStateMonitor</b> Used to monitor an integer or float value. When the value changes, the caption and color of this object change. To use this object, place it on screen and then add two or more OptoStateMonitorState objects. See this object’s online help for detailed configuration information.</p>	 <p><b>OptoStateMonitorState</b> Helper object for the OptoStateMonitor object. To use the OptoStateMonitorState object, place one object within the OptoStateMonitor object for each state you want to represent. See this object’s online help for detailed configuration information.</p>
 <p><b>OptoTextRadioButton</b> Displays a radio button graphic that represents a Boolean value or digital point on the Opto 22 device. If this object’s readonly property is set as false, then touching the object toggles the Boolean value/digital point. Touching this object also turns on or off all other objects of the same type in the same screen container.</p>	 <p><b>OptoTextToggleButton</b> Displays a checkbox graphic that represents a Boolean value or digital point on the Opto 22 device. If this object’s readonly property is set as false, then touching it toggles the Boolean value/digital point.</p>
 <p><b>OptoTrendChart</b> Displays a line graph that can be used to chart data. A memory map address on the Opto 22 device is queried periodically to update the chart. You can optionally have the chart log data when it (or its parent) is disabled, or have it start a fresh trend each time it is enabled to conserve system resources.</p>	

# Troubleshooting

## Solutions to Common Problems

If you have problems using your OptoTerminal, check the table below for common problems and steps to fix them. If the problem continues after trying the solutions listed below, contact Opto 22 Product Support for assistance. See [“For Help” on page 3](#) for contact information.

Problem	Solution Steps
<b>OptoTerminal has no power.</b>	<ol style="list-style-type: none"><li>1. Confirm that the cable is firmly connected to the terminal's DB9 female connector.</li><li>2. Confirm that the DB9 male connector connected to the terminal provides power to the correct connector pins. See the <i>OptoTerminal Hardware User's Manual</i> for connector pinouts and power requirements for the OptoTerminal-G70 and -G75.</li></ol>
<b>Qlarity Foundry application running on OptoTerminal is erased when power is cycled.</b>	<p>To keep an application in memory when power to the OptoTerminal is cycled, you must set the terminal to Download mode in the Power On Setup menu.</p> <ol style="list-style-type: none"><li>1. Cycle power to the OptoTerminal and start it up in the Power On Setup menu as described in <a href="#">“Configuring an OptoTerminal” on page 4</a>.</li><li>2. In the Power On Setup menu, confirm that the OptoTerminal is in Download mode, <i>not</i> Develop mode. Change the mode if required.</li><li>3. Select Save Changes and Exit to reboot the OptoTerminal.</li></ol>
<b>OptoTerminal can't communicate with Opto 22 hardware.</b>	<ol style="list-style-type: none"><li>1. Confirm that the correct IP address of the Opto 22 hardware the OptoTerminal needs to communicate with is entered in the Properties window in the Opto 22 communications object you are using in the application. For the memory map object library, the Opto 22 communications object is named <i>Opto22Obj</i>. For the PAC Control object library, the communications object is named <i>OptoPACComm</i>.</li><li>2. See the user's guide for your Opto 22 hardware for specific troubleshooting information for problems with network communication.</li></ol>
<b>PC can't communicate with OptoTerminal.</b>	Check the network TCP/IP settings for the PC and OptoTerminal and confirm that both are on the same subnet of the Ethernet network.

Problem	Solution Steps
<b>Can't download Qlarity Foundry application to OptoTerminal.</b>	This problem is usually due to the OptoTerminal not being set to Develop or Download mode. For more information, see the earlier troubleshooting topic <a href="#">“Qlarity Foundry application running on OptoTerminal is erased when power is cycled.”</a>
<b>Values received from Opto 22 I/O system are incorrect.</b>	Check that the I/O points have been configured correctly (for example, range and scaling values for analog I/O points). This is done using PAC Manager or PAC Control software. See the user's guide for your Opto 22 I/O system for specific configuration instructions.
<b>Qlarity Foundry application runs slowly on OptoTerminal.</b>	<ol style="list-style-type: none"> <li>1. Check how many Opto 22 objects are used in the application and what sampling rates are used for each object. <i>The objects should not cumulatively create more than about 50 updates per second</i> (for example, 50 objects with 1000 ms. sampling rates, or 25 objects with 500 ms. sampling rates).</li> <li>2. Make sure that the Draw Caching option is enabled in the Power On Setup menu.</li> <li>3. If draw caching is not enabled, check if multiple Opto 22 objects are stacked on top of one another. Delete stacked objects or change their positions.</li> </ol>
<b>Selected font does not appear on-screen, or you can't change the font on an Opto 22 library object.</b>	Check if a TrueType font is being used with an Opto 22 object. If so, verify that the TrueType is available.

## Updating OptoTerminal Firmware

For many operational problems, a good first course of action is to confirm that the terminal's firmware is current. See Chapter 7—Download Software in the *OptoTerminal Qlarity Foundry User's Manual* for instructions on how to check the version of firmware currently in the terminal and how to install new firmware if needed.

**Current firmware is included on the OptoTerminal CD, or you can download it from the Opto 22 Web site at [www.opto22.com](http://www.opto22.com).**

## Reviewing Diagnostic Information

To see useful diagnostic information about the application running on the terminal and other parameters, connect the OptoTerminal's serial port to the PC running Qlarity Foundry. This diagnostic information appears in Qlarity Foundry's Download Application dialog box.