SNAP ULTIMATE I/O LEARNING CENTER GUIDE

Form 1408-060105 (January 2006)

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

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017

email: support@opto22.com

SNAP Ultimate I/O Learning Center
Form 1408-060105 (January 2006)

Copyright © 2003–2006 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, OptoDisplay, OptoENETSniff, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC, 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

Getting Started

ioManager: Establishing Communication between the I/O and a PC

ioControl: Configuration and Programming

ioControl: Flowcharts

Digital Control

Analog Control

OptoScript

Operator Interfaces

Trending

Alarming

OptoOPCServer

Road Map

GETTING STARTED

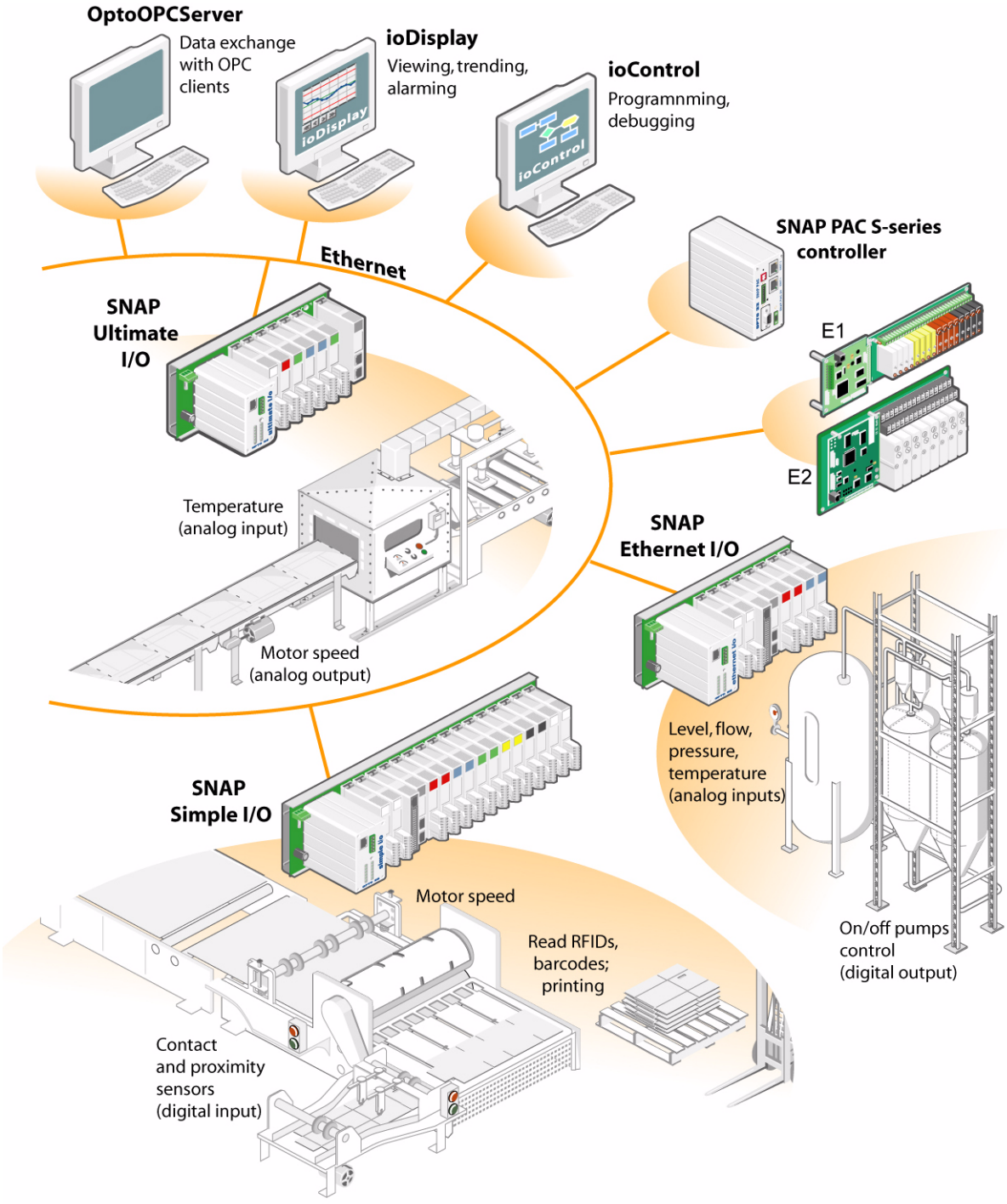
OPTO 22 AND SNAP PRODUCTS

Founded in 1974, Opto 22 develops and manufactures hardware and software products for applications in industrial automation, remote monitoring, and enterprise data acquisition. Using standard, commercially available Internet, networking, and computer technologies, Opto 22's input/output and control systems allow customers to monitor, control, and acquire data from all of the mechanical, electrical, and electronic assets that are key to their business operations. Opto 22's products and services support automation end users, OEMs, and information technology and operations personnel. More than 85 million devices worldwide are reliably connected to Opto 22 systems.

Opto 22's SNAP product family provides systems for remote monitoring, data acquisition, and industrial automation. The choice of system varies according to application needs, specifically, the level of control and the types of network and communication protocols.

SNAP Products Overview

Opto 22 provides a diverse set of products used in industrial automation, remote monitoring, and enterprise data acquisition.



I/O Unit

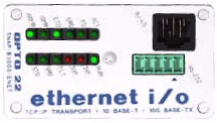
The SNAP Ultimate I/O system is essentially an I/O unit that is programmable using Opto 22 ioControl. The I/O system used in this Learning Center is just one I/O configuration available with the SNAP product line. The following shows the versatile SNAP systems you can build using racks, brains, and I/O modules.

SNAP I/O Processor (Brain)

SNAP Ultimate I/O™

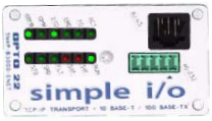
- Combination I/O processor and control engine running ioProject™ software
- Mixed I/O signal types (analog, digital, and serial)
- High-density digital-only version available

SNAP Ethernet I/O™

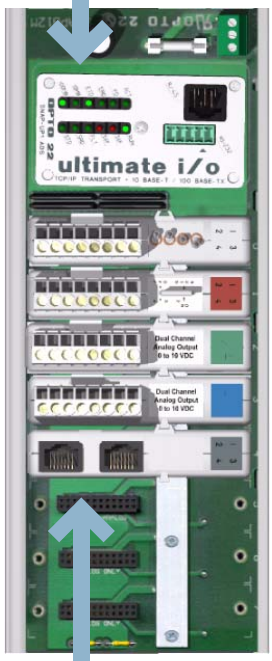


- Mixed I/O signal types (analog, digital, and serial)
- High-density digital-only version available
- Versions for wired and wireless networks
- Use with ioProject or FactoryFloor® controllers

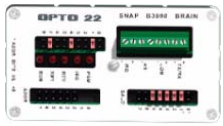
SNAP Simple I/O™



- Low-cost, high-quality system supporting analog, simple digital, and serial capabilities
- Use with ioProject or FactoryFloor family of controllers



SNAP Serial I/O



- High-performance I/O processor
- Analog and digital signal types
- Versions for RS-485, high-speed ARCNET, and Modbus
- Use with FactoryFloor family of controllers

SNAP Mounting Rack

- Rack sizes accommodate 4, 8, 12, or 16 modules for up to 64 points[1].
- Mix analog, digital, and serial modules on one rack.
- Rack holds one brain and up to 16 I/O modules.
- Each module connects two or four devices, depending on module type.
- Use DIN-rail or panel mounting.

SNAP I/O Modules	I/O Module Types	Available with SNAP Ultimate I/O, SNAP Ethernet I/O, and SNAP Simple I/O	Available with SNAP Serial I/O
	Analog Input[2]	2- and 4-channel	2-channel
	Analog Output	2-channel	2-channel
	Digital Input	4-channel	4-channel
	Digital Output	4-channel	4-channel
	Serial	2-channel	(Serial devices connect to the control engine)
	High-density Digital	32-channel	32-channel

1 The 64-point maximum per rack applies to standard modules. High-density digital modules provide 32 points per module; a maximum of eight of these modules on one rack provides a total of 256 digital points.

2 Channel-to-channel isolated versions are available.

SNAP Ethernet Systems Software

Opto 22 has developed ioProject, an integrated set of software and utilities for use with SNAP Ethernet systems. Collectively, these tools provide a simple yet powerful way to configure, design, and troubleshoot your control environments.

ioControl™

ioControl is a graphical, flowchart-based programming tool for machine control and process applications. Using ioControl, you create, download, and run control programs on a SNAP-LCE controller or on a SNAP Ultimate I/O unit. In addition to flowchart programming, ioControl includes a powerful, built-in scripting language based on C and other procedural languages.

ioDisplay™

ioDisplay is an intuitive HMI package for building operator interfaces for your Microsoft® Windows®-based clients communicating with Opto 22 SNAP Ethernet systems. ioDisplay offers a full-featured HMI including alarming, trending, security, and a built-in library of 3,000 industrial automation graphics.

OptoOPCServer™

OptoOPCServer is a fast, efficient OLE for Process Control (OPC) 2.0-compliant server that provides OPC clients access to SNAP Ethernet systems. Client applications can read and write to I/O points or even change their specific configuration options. OPC clients also have access to all strategy variables and elements in an ioControl strategy.

ioManager™

ioManager is a maintenance and communications utility provided with SNAP Ethernet systems hardware. This utility sets up the Ethernet communications, upgrades firmware, and streamlines the configuration of the I/O unit.

SNAP Ultimate I/O Learning Center Guide

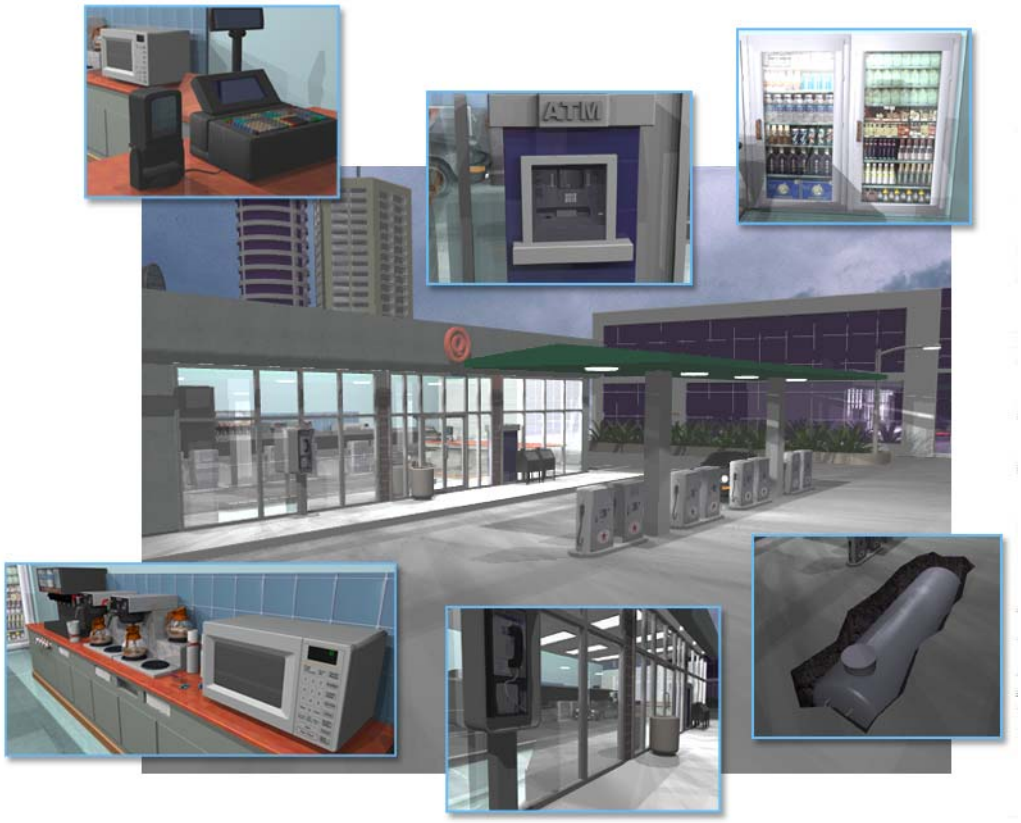
This *SNAP Ultimate I/O Learning Center Guide* is a hands-on, learn-by-example training course. Each chapter introduces aspects of ioControl, ioDisplay, OptoOPCServer, and ioManager. This training manual contains a step-by-step guide for each lesson.

What You Will Do

- Introduction: Assemble the Learning Center hardware.
- Lesson 1: Establish network communication with your Ultimate I/O systems using ioManager software.
- Lessons 2–6: Configure the I/O and program the SNAP Ultimate brain using ioControl.
- Lessons 7–9: Create an operator interface using ioDisplay.
- Lesson 10: Configure OptoOPCServer, which lets your SNAP Ultimate I/O systems connect with OPC-compatible software applications.

Learning Scenario: Convenience Store

Lessons 1–10 present their concepts within the context of a fictional scenario: using SNAP Ultimate I/O to remotely control and monitor a convenience store. SNAP Ultimate I/O is ideal for many types of applications, but a convenience store example allows us to demonstrate the rich set of features that combines aspects of machine control, monitoring, and data acquisition.



In this example, you will learn how to connect and control machinery, program logic using simple flowcharts and user-friendly project management, and create sophisticated user interfaces. These activities will illustrate how you can control many types of processes or monitor any type of machinery, and create complex process visualization programs.

Lesson Format

The following sections are used to assist you in learning, reviewing, and applying the material:

- **Skills**—an overview of skills taught in the lesson.
- **Scenario**—a description of how the lesson affects the convenience store application being built throughout the course.
- **Concepts**—an overview of the background concepts being applied in this lesson.
- **Activity**—step-by-step procedures and explanations for building the convenience store application, interspersed with concept reviews. Instructions within the

activity should be followed exactly. In some instances, upper or lowercase letters must be used. Type variable and I/O point names exactly as shown in this manual.

Project Road Map—the Road Map provides an overview of all the features covered in this Learning Center Guide. This may help you keep track of your progress and know where to review. It also shows you which features affect later chapters. For example, the Road Map shows you which points configured in previous chapters are used in later chapters.

Lesson Conventions

The following conventions are used throughout this guide:

Menu commands: Select *File* ⟶ *New* to create a new file.

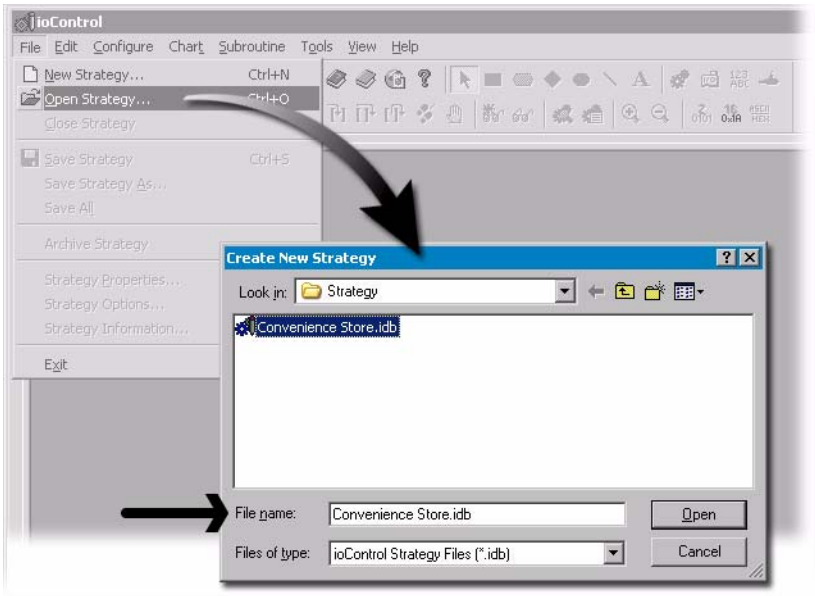
Computer files: Open the file **Store_IOConfig.otg**.

Text that you are to type: `Fuel is low`

Named selections on the screen: In the Type field, choose *Analog Point*.

Graphic illustrations of steps:

- Curved arrows show preceding step(s).
- Straight arrows emphasize a feature of interest.



This diagram is an example of the type of instructions used throughout this guide.

SNAP Ultimate I/O Learning Center Supplements

When you've completed the activities in this *SNAP Ultimate I/O Learning Center Guide*, there is still more to discover. Learning Center Supplements (OptoTutorials™) provide step-by-step instructions for advanced features of ioProject software. To download the latest OptoTutorials, visit www.opto22.com/site/le_index.aspx.

WHAT YOU NEED

System Requirements

To use the SNAP Ultimate I/O Learning Center with your PC, you must have the following minimum computer configuration for ioProject software:

- A computer with at least the minimum processor required for your version of Microsoft Windows (1 GHz Pentium-class or better recommended) with an Ethernet card.
- VGA or higher resolution monitor (Super VGA recommended). Minimum size: 800x600 with small fonts
- Mouse or other pointing device
- Installed Windows printer (optional)
- Microsoft Windows XP (with service pack SP1) or Windows 2000® (with SP4) workstation operating system
- At least 128 MB RAM (256 MB recommended)
- If you are using ioDisplay and your ioDisplay project uses many basic trends, SuperTrends, or XY Plots, additional RAM is strongly recommended. See Opto 22 form #1302, the *ioDisplay User's Guide*, for more information.
- Available hard disk space:

ioControl	15 MB
ioDisplay	50 MB
OptoOPCServer	50 MB
ioManager	3 MB

Special Note on Windows XP Service Pack 2

Windows XP service pack SP2 was released too late in our development cycle to test it with ioProject 6.0 software. If you are using SP2 with ioDisplay or OptoOPCServer, you should read the following OPC Foundation paper on how to configure Windows XP so that OPC and DCOM will work properly:

www.opcfoundation.org/WebUI/DownloadFile.aspx?CM=1&RI=161

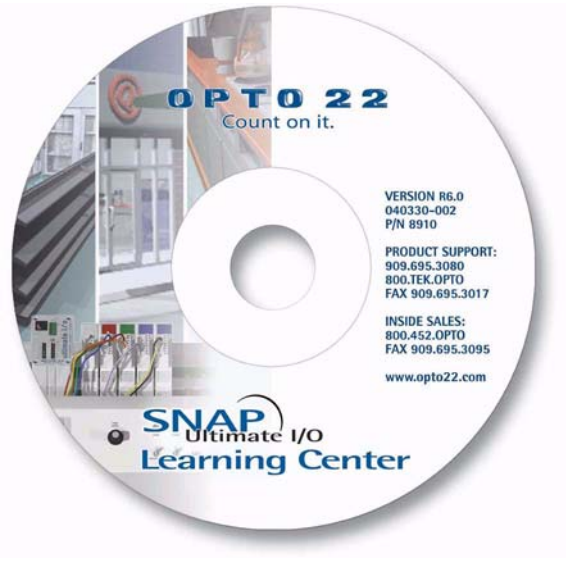
Installing the SNAP Ultimate I/O Learning Center Software

The SNAP Ultimate I/O Learning Center includes all the software you need to get hands-on experience with SNAP Ultimate I/O from Opto 22. Installation of the SNAP Ultimate controller/brain software is easy and quick.

Remove the SNAP Ultimate I/O Learning Center CD-ROM from the Opto 22 CD wallet and insert it into your computer's CD drive. (NOTE: Several CDs are included with the Learning Center. Make sure you find the one that appears as shown here.)

The installation wizard should appear. If the wizard does not appear, start Windows Explorer, navigate to your compact disc drive, and double-click Setup.exe to begin installation.

If you have trouble installing the software, contact Opto 22 Product Support. See "Opto 22 Product Support" on page xxii.



The SNAP Ultimate I/O Learning Center CD is found in the Opto 22 CD wallet

The SNAP Ultimate I/O Learning Center CD includes:

- A complete version of ioProject software and utilities
- Training sample files, which are installed in **C:\Program Files\Opto22\ioProject\UIO Learning Center**

Setting up the SNAP Ultimate I/O Learning Center

The SNAP Ultimate I/O Learning Center includes all the necessary Opto 22 I/O hardware, a load panel with power supply, and all necessary cables.

SNAP Hardware Components

SNAP Ultimate Controller/Brain

The SNAP Ultimate controller/brain is one of Opto 22's high-performance processors with an on-board control engine designed to control analog, digital, and serial modules on the same rack.

The SNAP Ultimate controller/brain merges two functions that are usually located in separate pieces of hardware: input/output processing and flowchart-based control. The brain, rack, and modules together are referred to as an *I/O unit*. You can attach the I/O unit to existing Ethernet networks, making it easy to add control where necessary. Or you can use standard Ethernet connections to build an independent control network, connecting your PC directly to the I/O unit.

SNAP-B8M Rack

The SNAP-B8M rack mounts up to eight SNAP digital, analog, and serial modules.

SNAP I/O Modules

The Learning Center includes the following SNAP I/O modules:

- SNAP-IDC5D: Digital, DC Input 2.5–28 VDC, 5 VDC Logic
- SNAP-ODC5SRC: Digital, DC Output 5–60 VDC Source, 5 VDC Logic
- SNAP-AOV27: Analog Output, Dual -10 to +10 VDC
- SNAP-AICTD: Analog Input, Dual ICTD Temperature
- SNAP-AIV: Analog Input, Dual ±10 or ±5 VDC (configurable)

Learning Center Load Panel

The load panel consists of:

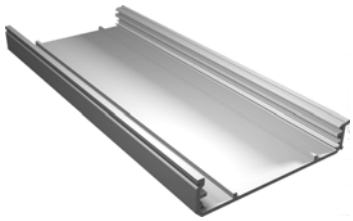
- Two toggle switches
- Two momentary switches
- One SonAlert (buzzer)
- Three LEDs
- One meter
- One potentiometer
- Temperature sensor connection

Accessories

The SNAP Ultimate I/O Learning Center includes the following accessories:

- Temperature probe
- Power cable
- Ethernet Category 5 crossover cable—This cable allows a direct connection between your PC and the SNAP Ultimate I/O unit without the need for a hub or additional cables.

Learning Center hardware components are illustrated below:



SNAP rack extrusion

DIN-rail clip with spacing tab and slider (sample)

SNAP-B8M rack

Learning Center load panel

SNAP Ultimate controller/brain

SNAP I/O module (sample)

Temperature probe

SNAP module tool

Ethernet crossover cable

Required Tools

Phillips and flatblade screwdrivers are provided for assembling the SNAP Learning Center.

ASSEMBLING THE SNAP ULTIMATE I/O LEARNING CENTER

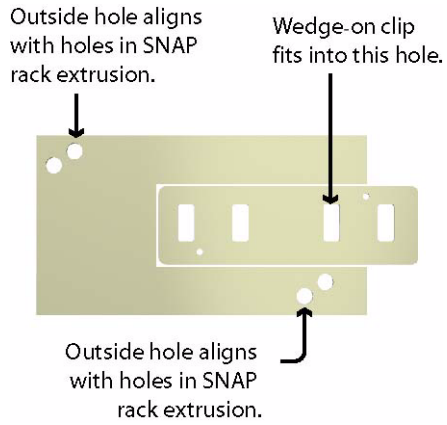
Follow the instructions in each section below to assemble the components of your SNAP Ultimate I/O Learning Center and connect it to your PC.

Installing the SNAP-B8M Rack

1. Align the parts of the DIN-rail clips.

The diagram to the right shows how to position the two parts of the DIN-rail release spacer. Place these two parts on top of the metal DIN-rail clip.

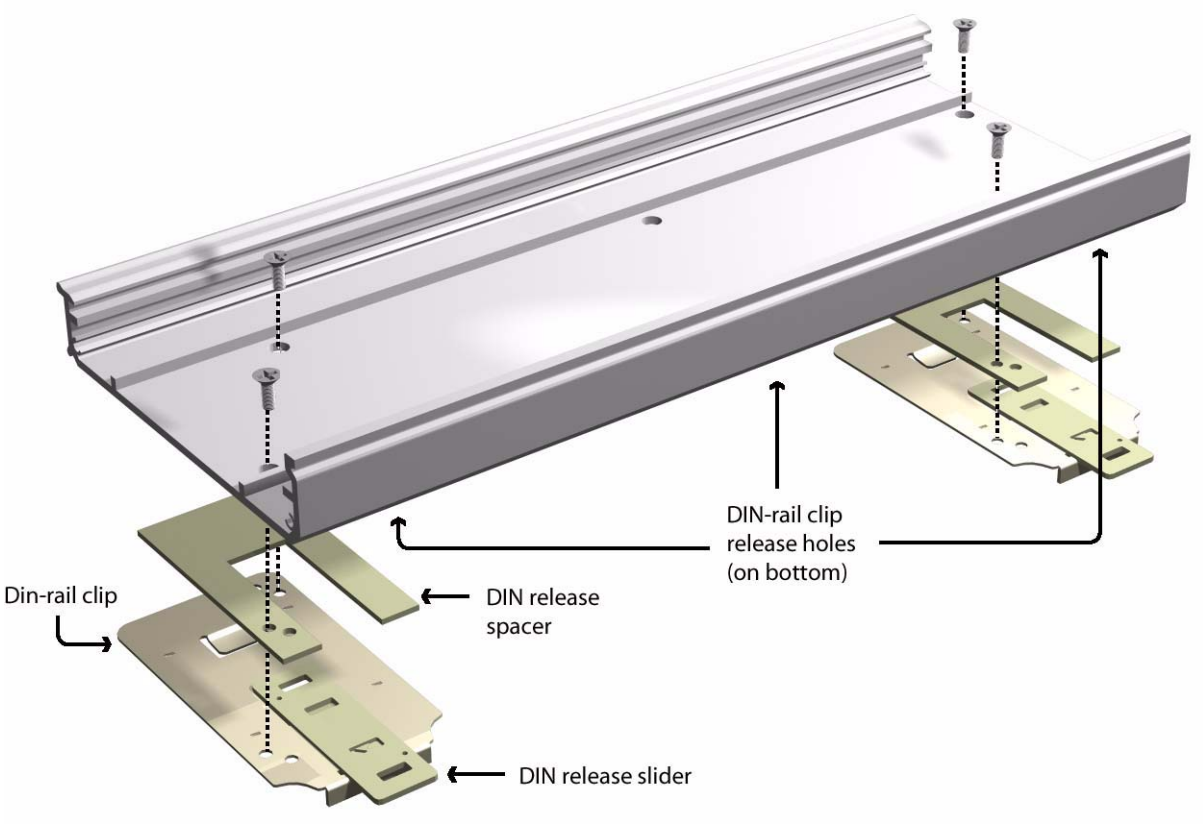
NOTE: Do not use the rectangular aluminum plates included with the DIN-rail clips.



2. Install the DIN-rail clips onto the SNAP rack extrusion as follows:

a. Place the plastic DIN release spacers between the DIN-rail clips and the extrusion as shown in the following diagram.

b. Align the screw holes on DIN-rail clips and spacers with the screw holes on the extrusion, and then install the screws.



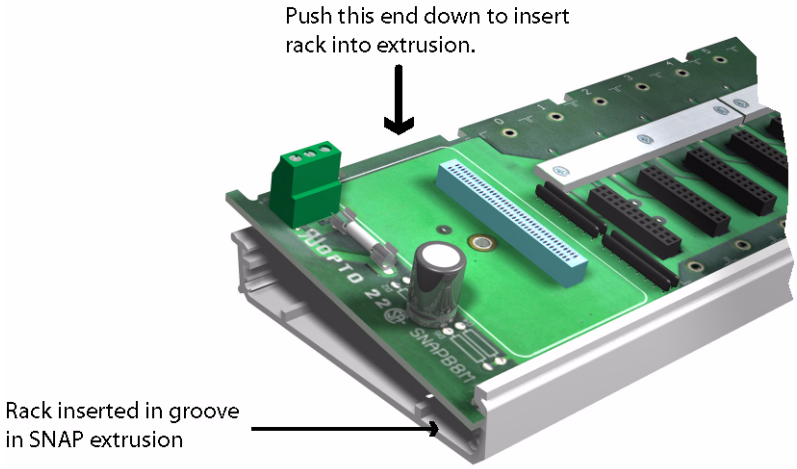
3. Install the SNAP rack extrusion and the DIN-rail clip assembly onto the DIN rail as follows:

a. Align the extrusion so that the taller lip is closest to the Learning Center load panel and the lower edge is towards the back. See the next illustration.
b. Hang the fixed end of each DIN-rail clip over the top edge of the DIN rail.
c. Position the extrusion and clip assembly flat against the DIN rail.

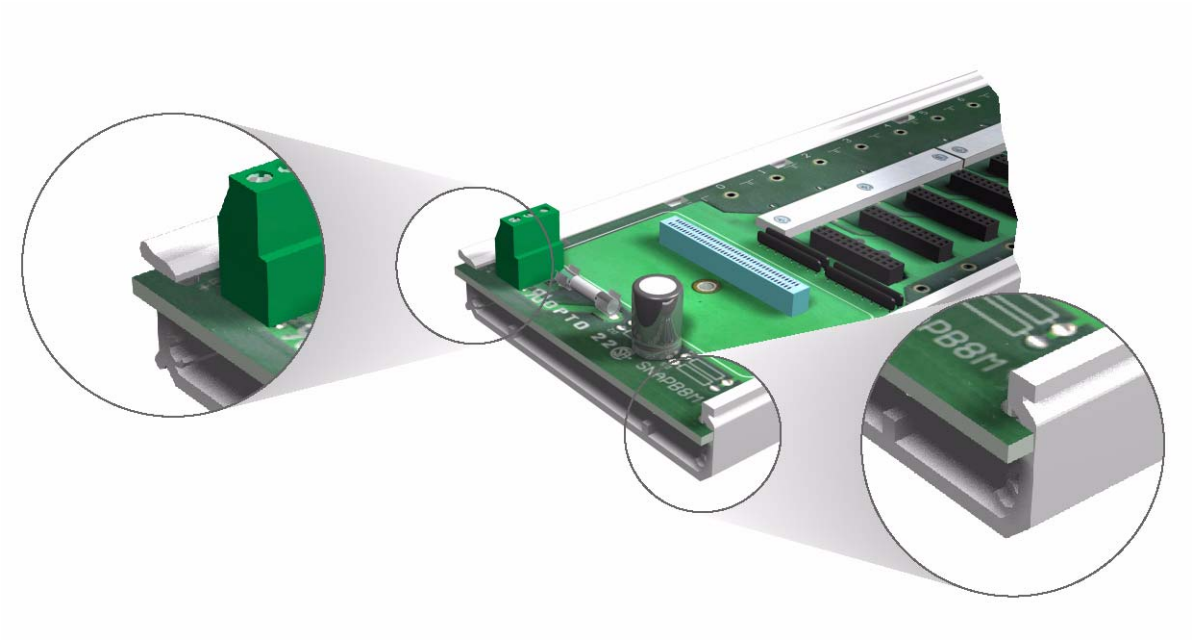
d. Apply pressure to the bottom of the extrusion and clip assembly to snap it into position on the Learning Center's DIN rail.

4. Install the SNAP rack in the rack extrusion as follows:

a. Orient the circuit board so that the module connectors are at the bottom and the circuit board release notches on the circuit board are at the top.
b. Align the bottom edge of the circuit board with the top groove in the extrusion above the DIN-rail clip release holes as shown in the illustration below.
c. Press the top of the circuit board against the extrusion and snap the board into the groove on the top side of the extrusion. (It's okay to push hard.)



A correctly inserted rack appears as shown here:



Connecting the Power Supply

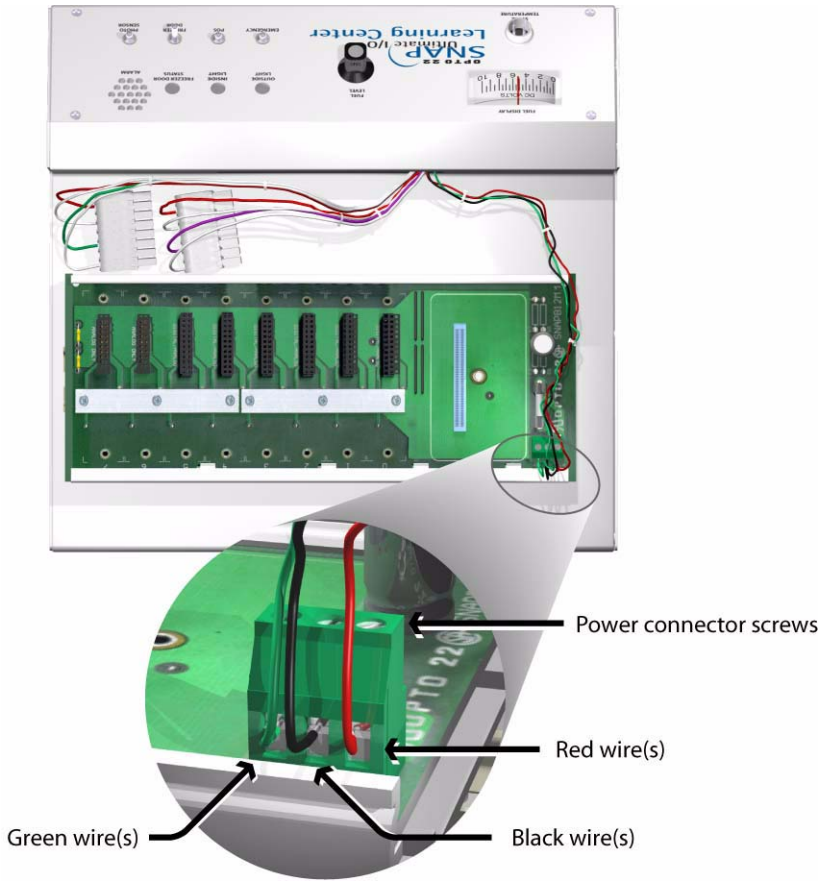
1. **Locate the wire harness behind the load panel of the Learning Center.**

2. **Locate the red, black, and green wires that have bare leads at one end.**

3. **On the SNAP-B8M rack, connect the red wire to the +5V connection and tighten its screw on the power connector as shown below.**

4. **Connect the black wire to the -5V connection and tighten its screw.**

5. **Connect the green wire to the ground connector and tighten its screw.**



Installing the SNAP Ultimate Controller/Brain

1. **Remove the SNAP Ultimate controller/brain from its packaging.**

2. **Install the brain on the SNAP rack as follows:**

   a. Align the brain connector with the mating connector on the SNAP rack as shown in the following illustration.
   b. Seat the brain onto the connector.
   c. Tighten the hold-down screw to secure the brain in position.

CAUTION: Do not overtighten the hold-down screw. Doing so may damage the connector, the brain, or other components on the SNAP rack.

Installing the SNAP I/O Modules

1. **Remove the five SNAP I/O modules from their packaging.**

2. **Remove the screw terminal connectors from the tops of the modules and set them aside.**

   You will not use the terminal connectors that came with the modules, as the Learning Center has already been wired for you.

3. **Install the modules on the SNAP rack in the order shown in the table below.**

Module Positions

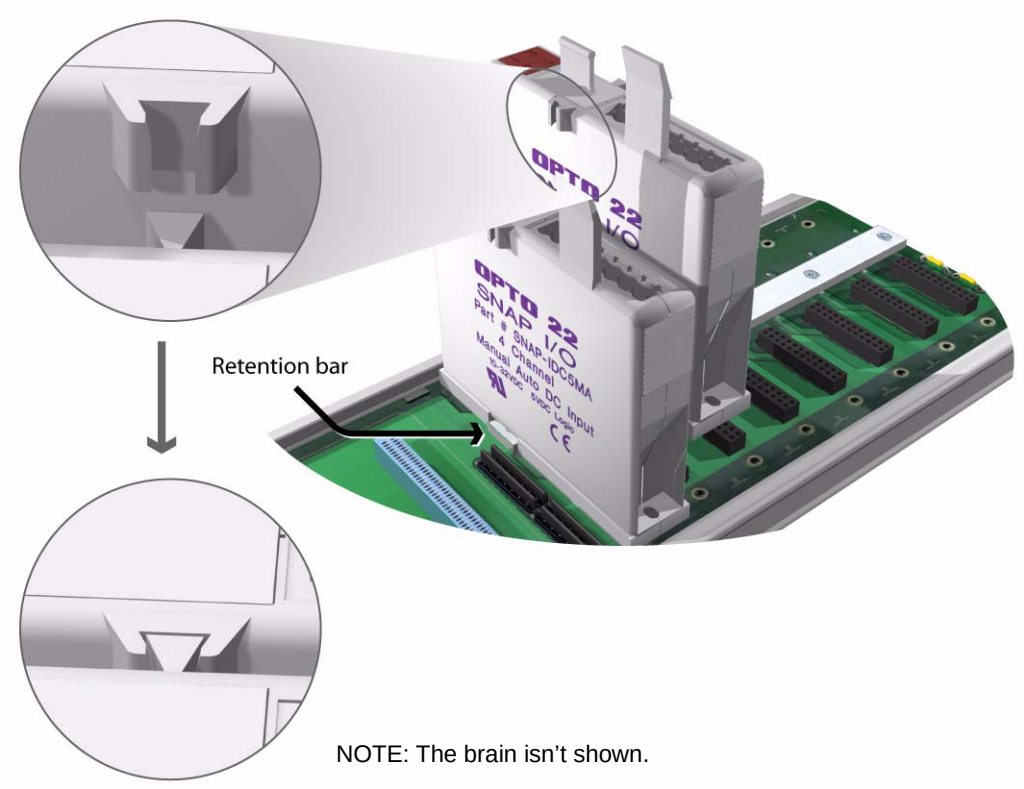
SNAP Module	Position on SNAP Rack
SNAP-IDC5D	0
SNAP-ODC5SRC	1
SNAP-AOV-27	2
SNAP-AICTD	3
SNAP-AIV	4

Module positions are labeled on the SNAP rack. Also, the modules are shown in order from left to right as you face the load panel.

To install each SNAP I/O module on the SNAP rack, follow these steps:

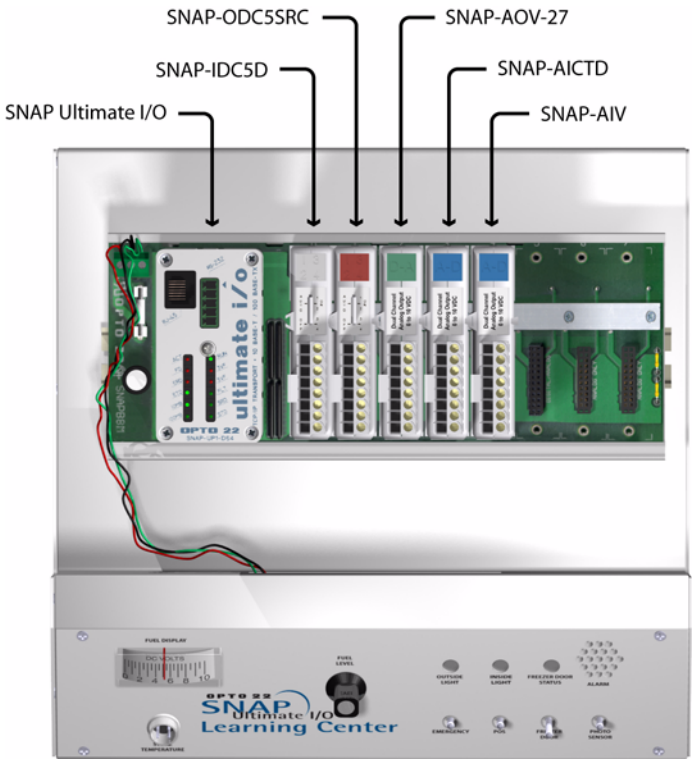
a. Position the module over the connector, aligning the small slot at the base of the module with the retention bar on the rack.
b. With the module correctly aligned over the connector, push straight down on the module to snap it into position.

When positioning modules next to each other, be sure to align the male and female module keys (shown in the detailed view in the illustration below) before snapping a module into position.



NOTE: The brain isn't shown.

Your Learning Center should now resemble the illustration below. If a module is positioned incorrectly, remove it (See “Removing a SNAP I/O Module” on page xvii.), and then reinstall it.



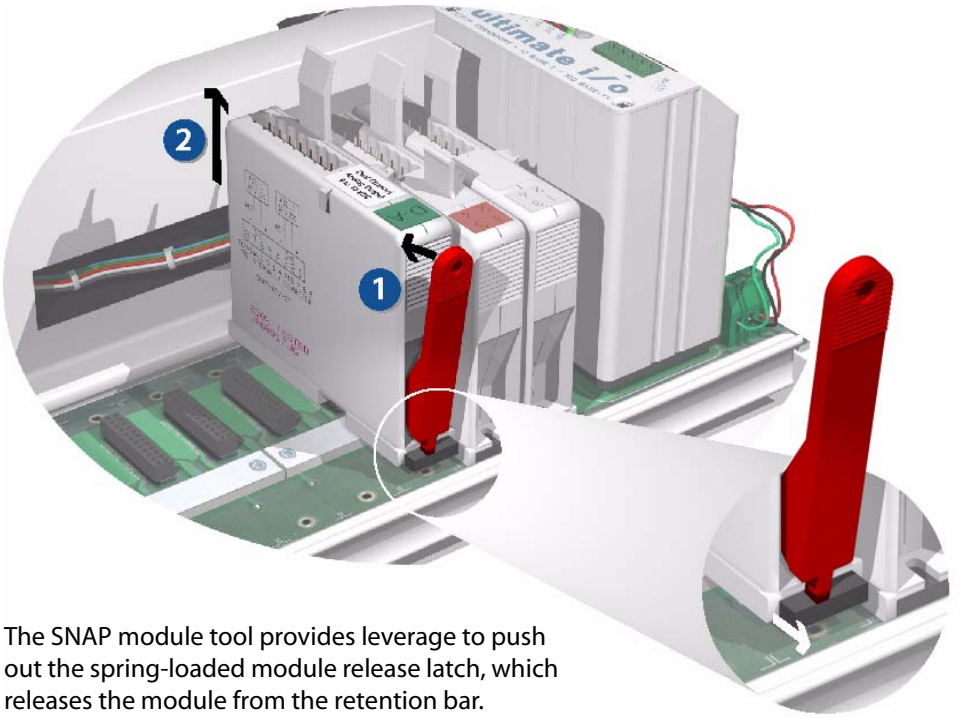
Removing a SNAP I/O Module

If you insert a module in the wrong location, you can remove it as follows:

1. **Use the SNAP module tool shown in the next illustration to retract and hold the release latch at the base of the module.**

   Note that the release latch is located on the end of the module facing the back of the Learning Center.

2. While holding open the release latch with the SNAP module tool, pull straight up on the module to remove it.

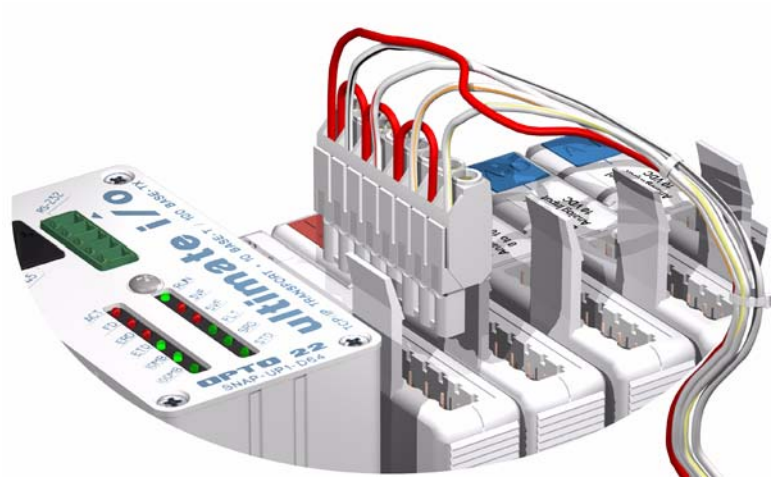


The SNAP module tool provides leverage to push out the spring-loaded module release latch, which releases the module from the retention bar.

Connecting the SNAP Modules to the Wiring Harness

1. Snap the screw terminal connectors on the Learning Center's wiring harness into the SNAP modules in the following order:

a. Plug the first connector to branch off from the wiring harness (after the power wires) into the first module, the SNAP-IDC5D.



b. Plug subsequent connectors that branch off the wiring harness into the remaining modules. Note that the connectors are on different lengths of wire to help indicate which connector goes with each module. (The shortest wire corresponds to slot 0.)

2. Carefully double-check your connections by comparing them with the following list of SNAP modules and the corresponding wire.

Wiring SNAP Modules

SNAP Module	Wire colors for corresponding connector on wiring harness
SNAP-IDC5D	red, blue/white, red/white, orange/white, yellow/white
SNAP-ODC5SRC	red, red, green/white, blue/white, purple/white, gray/white
SNAP-AOV-27	orange, yellow, green, blue
SNAP-AICTD	gray, purple
SNAP-AIV	white, black

Connecting the Temperature Sensor

The temperature sensor is the cable approximately 3 feet (1 meter) in length with a black plastic tip at one end.

1. Plug the temperature sensor into the jack on the front left-hand side of the Learning Center load panel.

NOTE: The four small prongs in the plug must align with the holes in the jack.

2. Tighten the threaded collar.

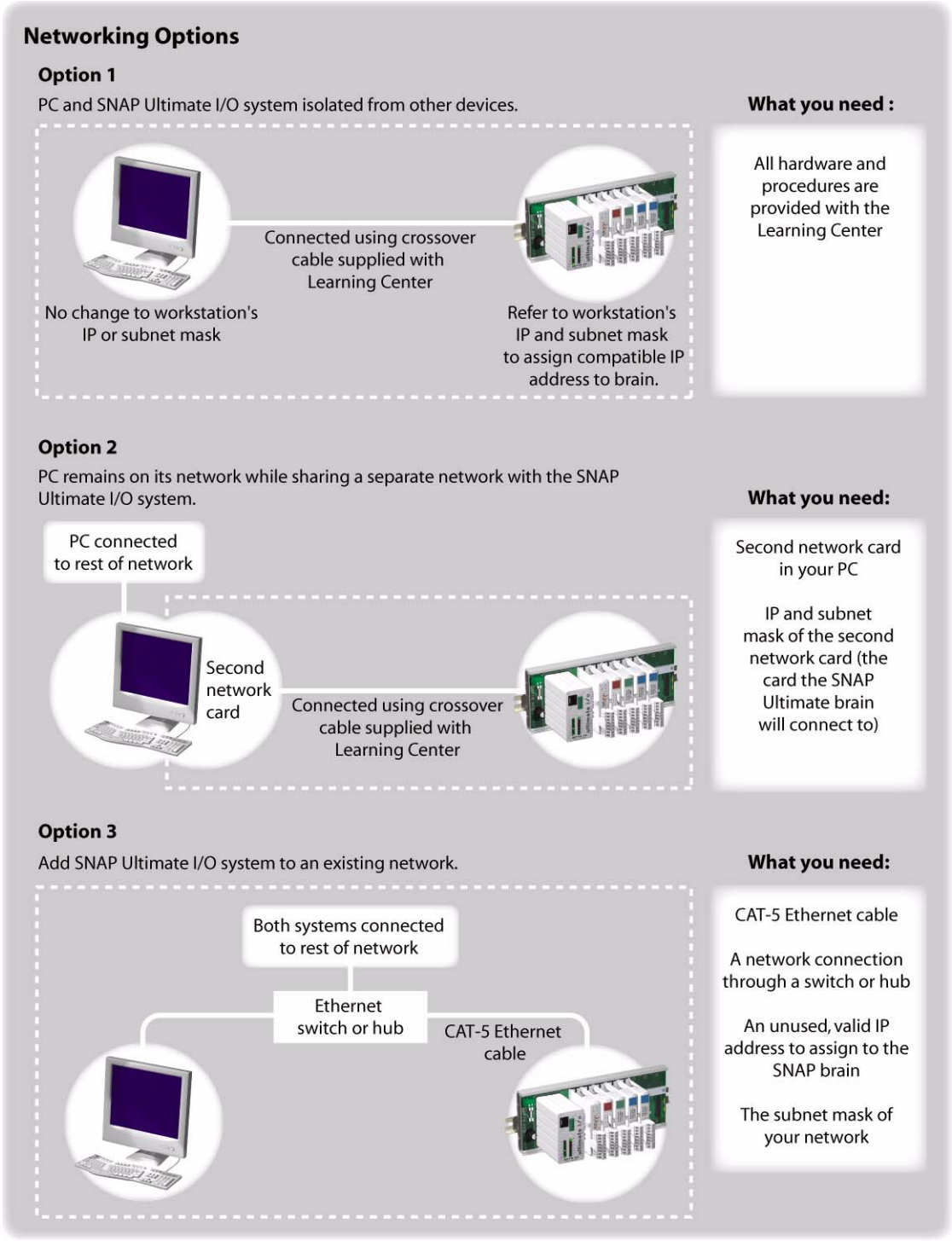
Connecting the Power Cable

To connect the power cable to the SNAP Ultimate I/O Learning Center, plug the cable into the power cord connector located on the right side of the Learning Center base.

NOTE: Don't turn on the Learning Center until instructed to do so in the next chapter.

Connecting the SNAP Ultimate I/O Unit to a PC

The SNAP Ultimate I/O unit requires an Ethernet connection to the PC running the ioProject software. There are three options to connect, as described below.



The Learning Center provides all the hardware to connect using Option 1; however, it my not be desirable to remove your PC from an existing network. Options 2 and 3

preserve your PC's connection with a network, but require some networking expertise. It is recommended that you consult with your network administrator on which method to use.

Connecting Directly

The simplest way of connecting the I/O unit and a computer is using the Ethernet crossover cable included with the Learning Center. This is shown in the illustration below.

IMPORTANT: An Ethernet crossover cable must be used for this direct connection. The PC and the I/O unit cannot communicate if a regular, or "straight-through," Ethernet cable is used.

1. **Plug one end of the blue Ethernet cable into the RJ-45 port on the SNAP Ultimate brain.**

2. **Plug the other end of the cable into an Ethernet port on the PC.**

If an Ethernet port isn't available because the computer is connected to a company network, consider one of the alternate methods of getting an available Ethernet port listed below:

- Disconnect the computer from the Ethernet network.

  IMPORTANT: Check with your network administrator before disconnecting the computer from the network.

- Install and configure a second Ethernet network card in the PC.

NOTE: This can be a complex task if you are not familiar with setting up a Windows PC. You may want to get assistance with installing the network card or have the installation and configuration done by your system administrator or others.

Once you have an Ethernet port available on the PC, connect the SNAP Ultimate I/Obrain as described above.

Connecting through a Network

The SNAP Ultimate I/O unit can be connected as a node on an existing Ethernet network. Once connected to the network, the SNAP Ultimate I/O unit can be accessed using the software included with the Learning Center.

When connecting the I/O unit to an existing Ethernet network, it is important to work closely with your network administrator, who must determine network topology and hardware requirements. The network administrator must understand that the I/O unit requires a **fixed** IP address. If a DHCP or BootP server is active on the network, the

system administrator will need to provide you with a static IP address for use with the brain. You will learn how to assign an IP address in Lesson 1.

Opto 22 Product Support

Opto 22 is here to help. When trying to solve a problem, a good rule of thumb is never spend more than an hour working on it before you call Opto 22. Your success is our success.

Phone: 800-TEK-OPTO (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 Web site: support.opto22.com

NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received; neither takes priority.

When calling for technical support, tell the Product Support Engineer you are using a SNAP Ultimate I/O Learning Center. Also be prepared to provide:

- your PC configuration (type of processor, speed, memory, and operating system)
- any specific error messages seen.

ioManager: Establishing Communication between the I/O and a PC

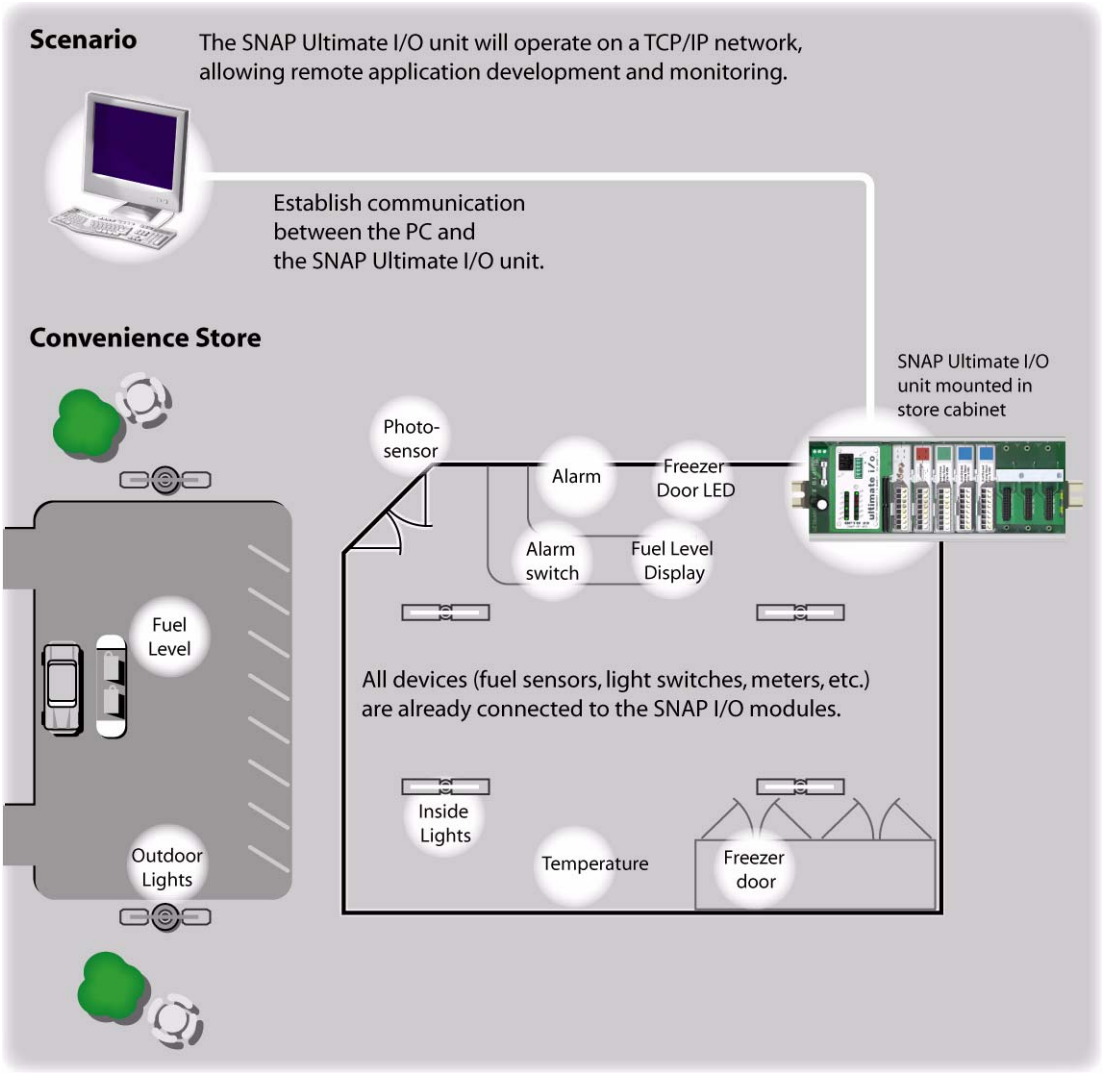
Skills

Network Communication

- Determining your PC's I/O Address and Subnet Mask
- Using ioManager to establish initial communication with the Brain
- Assigning network IP addresses using ioManager

Scenario

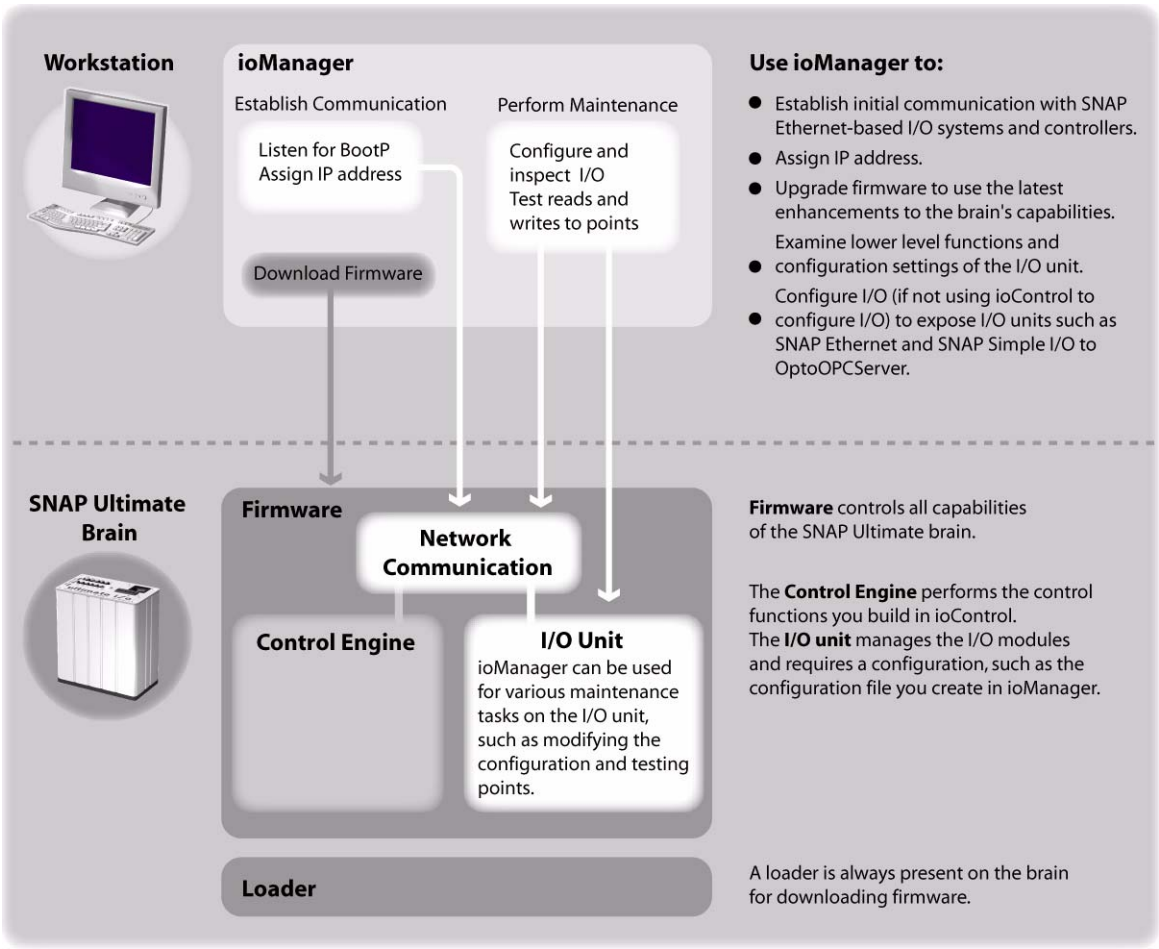
In this lesson, you begin deploying the SNAP Ultimate I/O system in a fictional convenience store represented by the instruments on the Learning Center's load panel. The scenario begins with a newly installed SNAP Ultimate I/O system for monitoring and controlling various devices in a convenience store. Your SNAP Ultimate I/O system is already wired to your field devices (Learning Center Load Panel) and is connected to its power supply. Your first task is to establish network communications with the I/O unit.


Scenario
The SNAP Ultimate I/O unit will operate on a TCP/IP network, allowing remote application development and monitoring.
Establish communication between the PC and the SNAP Ultimate I/O unit.
Convenience Store
SNAP Ultimate I/O unit mounted in store cabinet
Photo-sensor
Alarm
Freezer Door LED
Alarm switch
Fuel Level Display
Fuel Level
All devices (fuel sensors, light switches, meters, etc.) are already connected to the SNAP I/O modules.
Inside Lights
Temperature
Freezer door
Outdoor Lights

CONCEPTS

ioManager

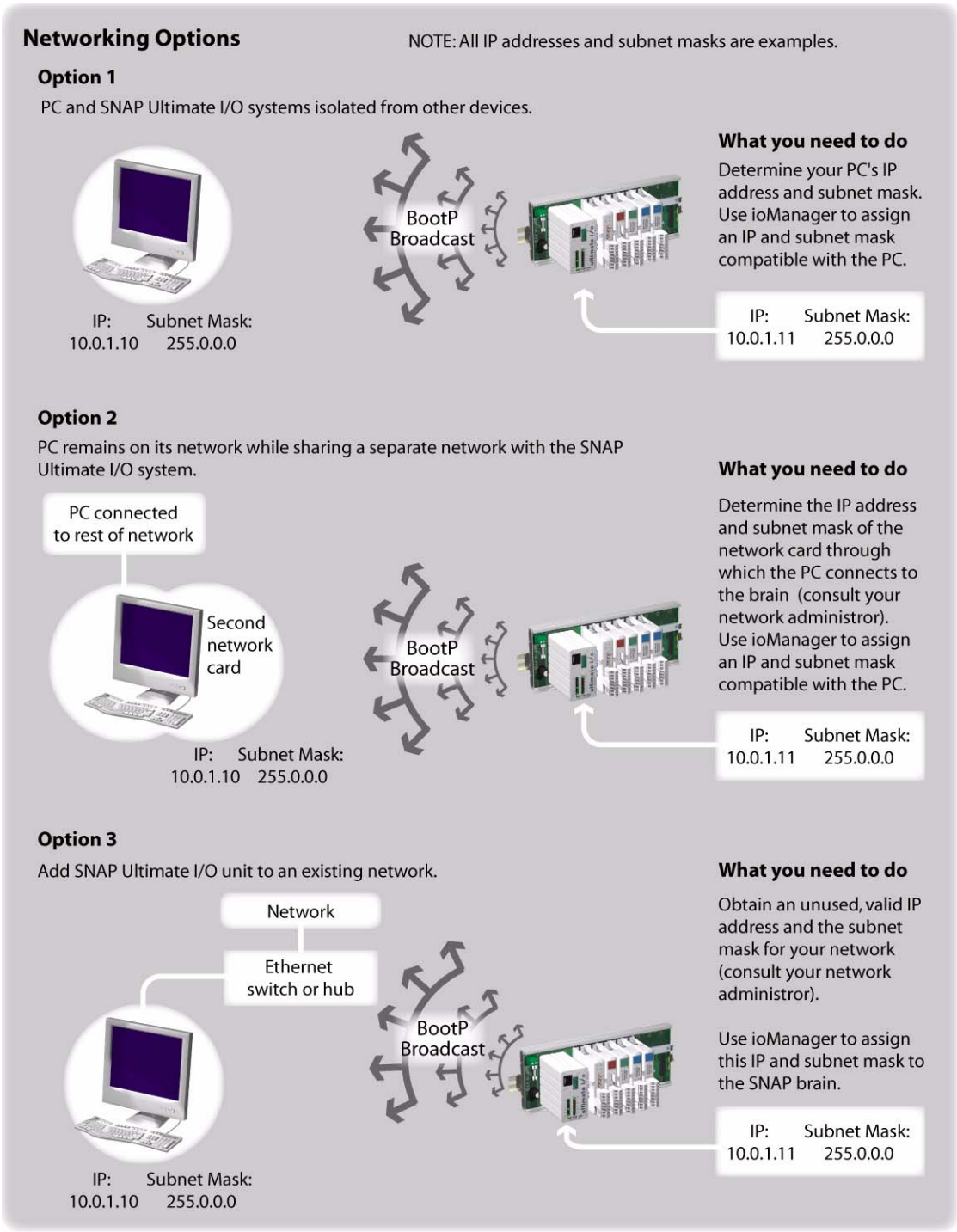
ioManager simplifies the setup of your SNAP Ultimate I/O unit by providing basic management features. The diagram below outlines ioManager's functions. In this lesson, you will use ioManager to establish communication with the I/O unit, assign an IP address, and update the SNAP Ultimate brain's firmware.



IP Addressing

The SNAP Ultimate controller/brain is shipped from the factory with a default IP address of 0.0.0.0. This address is not a valid network address. When you use your I/O unit for the first time, the brain will send a BootP network broadcast requesting an IP address. ioManager "hears" the BootP broadcast, and from ioManager, you can assign an IP address. To assign an IP address, you need an IP address and subnet mask that are valid for your network.

.



ACTIVITY

Preparing for the Lesson

You will need to know a valid IP address and subnet mask to assign to your SNAP brain. If your PC and brain are directly connected as described by Option 1 (above), this

activity tells you how to determine a valid IP address and subnet mask. If you are using Options 2 and 3, it is best to consult your network administrator for an IP address and subnet mask.

Determining Your PC's IP Address and Subnet Mask

Your SNAP brain will need the same subnet mask that is also assigned to your PC. The IP address will need to be compatible with the network your PC and SNAP brain are on. If you already have a valid IP address and subnet mask to assign to your SNAP brain (for example, you received these from your network administrator), you can skip the following steps and continue with "Using ioManager to Establish Communication with the Brain" on page 6.

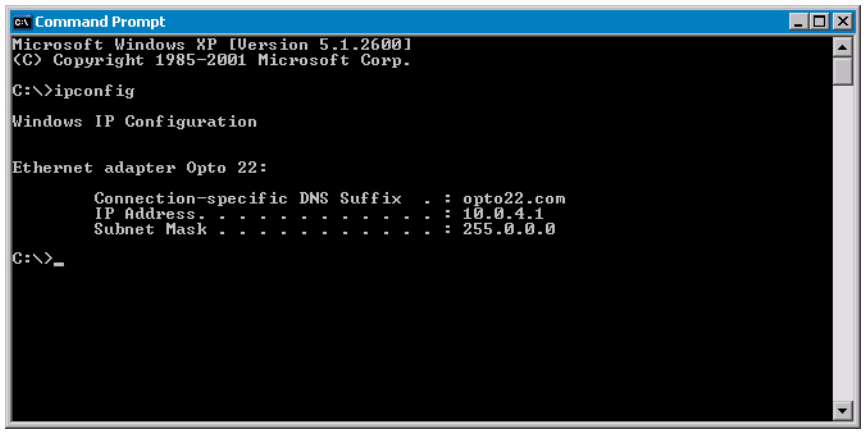
1. Open the MS-DOS Prompt.

Choose *Start* ⟶ *Programs* ⟶ *Command Prompt* or choose *Start* ⟶ *Run*, type `CMD` in the Run dialog box, and click *OK*.

2. Type `ipconfig` and press Enter.

Your computer's IP address and subnet mask are shown.

Choose a similar IP address for your I/O unit. For example, if your PC's IP address is 10.0.4.2, assign an IP address such as 10.0.4.1 or 10.0.4.3 to your Ultimate I/O.



3. Close your Command Prompt.

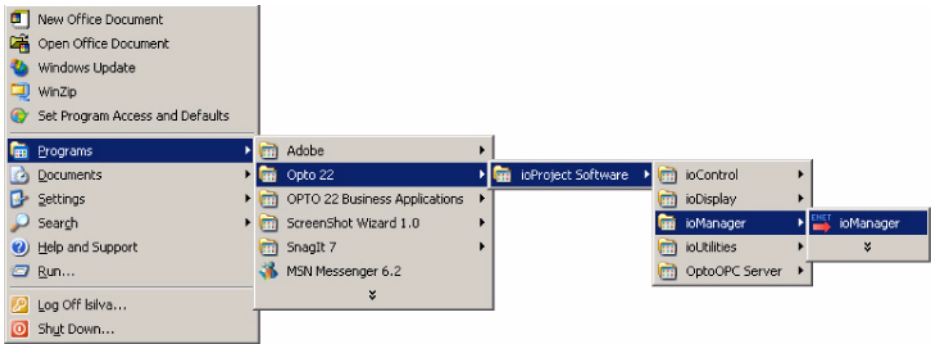
Click the close button in the top right-hand corner of the Command Prompt window.

Using ioManager to Establish Communication with the Brain

Once you've established what your brain's IP address and subnet mask should be, you can establish communication between your PC and the brain.

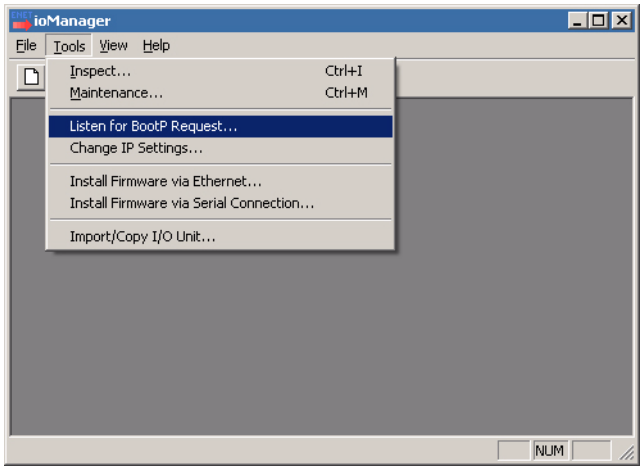
1. Launch ioManager.

a. From the Start menu, choose
Programs ⟶ *Opto 22* ⟶ *ioProject Software* ⟶ *ioManager* ⟶ *ioManager.*



2. Listen for BootP request.

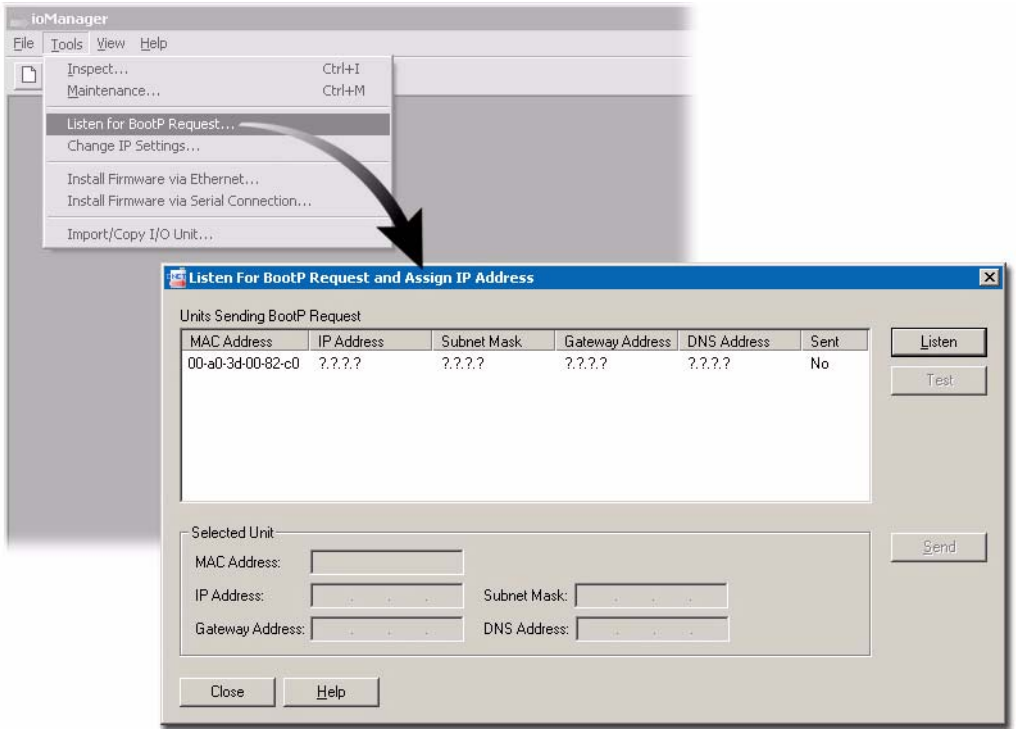
Select *Tools* ⟶ *Listen for BootP request.*



3. Turn on your Ultimate I/O Learning Center.

a. The power switch is next to the power cord connection.

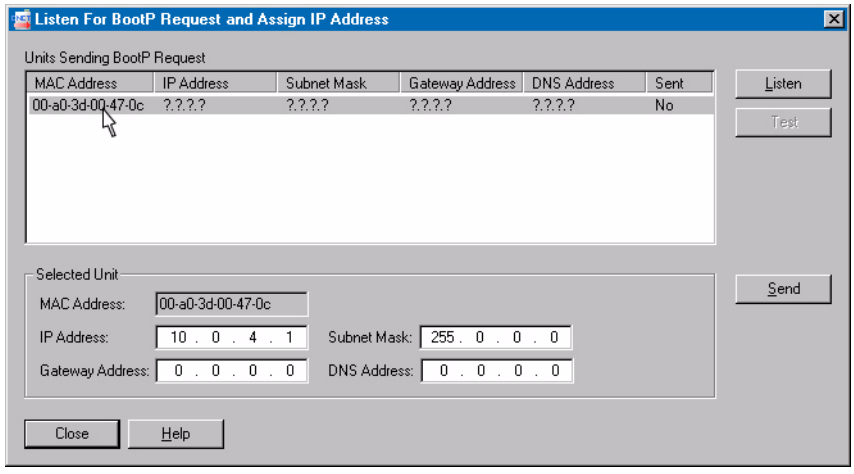
b. The brain is now broadcasting a BootP request for its network configuration.



Your Ultimate I/O unit is identified by its MAC address, also written on a label attached to the side of the brain's housing. (The MAC addresses of Opto 22 devices begin with 00.a0.3d.) If no units are listed in the Units Sending BootP Request, turn your brain off, click *Listen*, and turn your brain on.

4. Assign an IP address.

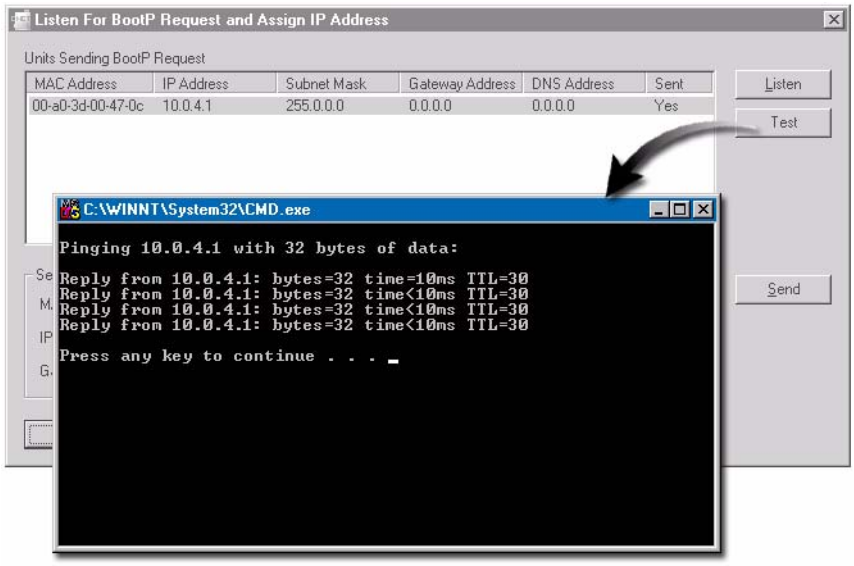
a. Click your brain's MAC address.



b. Type a suitable IP address (`10.0.4.1` is used in this training guide).
c. Type a suitable Subnet Mask (`255.0.0.0` is used in this training guide).
d. Click *Send*.
e. Click *OK* to close the ioManager confirmation message.

5. Test the IP Address.

a. Wait a few seconds for the new IP address to take effect.
b. Click *Test*.



The MS-DOS window indicates whether your brain is communicating with the network.

You should see a reply message similar to the one shown above. NOTE: If you encounter problems downloading or running your strategy, see the troubleshooting assistance in the ioControl User's Guide (Opto 22 form #1440). User documentation is on the SNAP Ultimate I/O Learning Center CD in the Manuals folder.

c. Press any key to close the MS-DOS window.

6. Close the Listen For BootP Request and Assign IP Address dialog box.

ioControl: Configuration and Programming

Skills

Building a Strategy

- Creating a new strategy
- Configuring a control engine
- Configuring I/O units and points
- Downloading a strategy

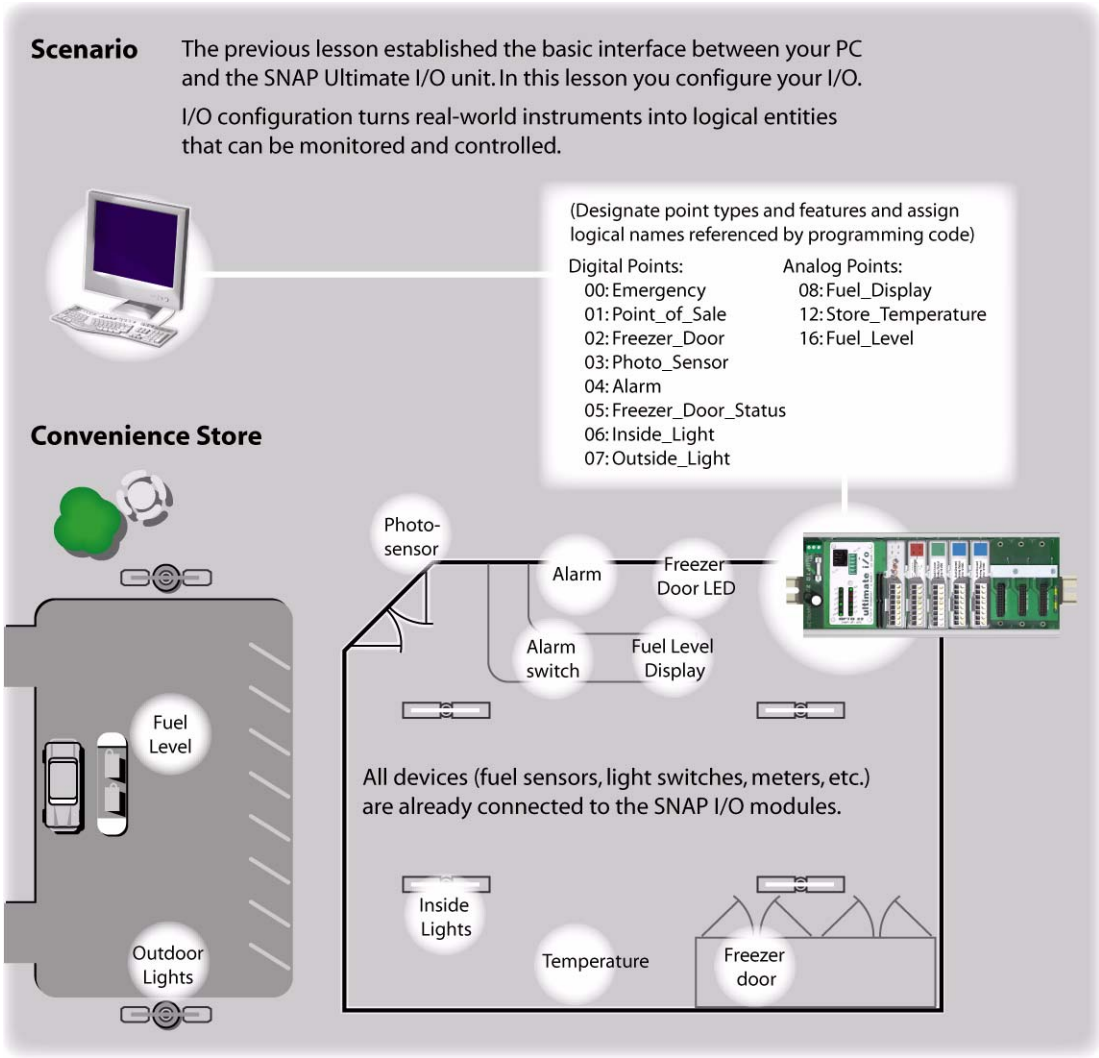
Debugging a Strategy

- Creating a New Strategy
- Configuring the Control Engine
- Configuring the I/O Unit
- Configuring I/O Points
- Making the Strategy Independent of Specific Controllers
- Observing Points in Debug Mode
- Writing to Outputs and Simulating Inputs in Debug Mode

Scenario

Using ioManager, you assigned your SNAP Ultimate I/O unit an IP address so that your PC can "see" it on the network. Now you are ready to build your control strategy. A strategy is the software program you create in ioControl. It includes all definitions and instructions necessary to control your process.

In this lesson, you will configure the SNAP Ultimate I/O unit so it can actuate and read sensors connected to your I/O modules.



CONCEPTS

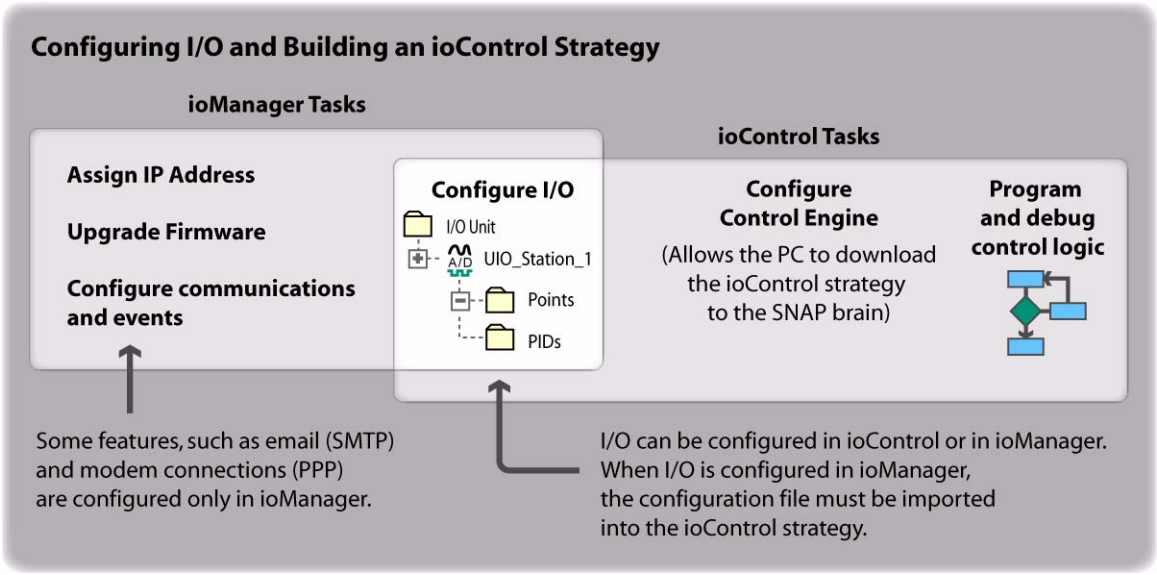
I/O Configuration

In the previous lesson, you used ioManager to establish network communication with the SNAP Ultimate I/O unit. The next step is to set up the basics of your strategy in ioControl, so you can assign a control engine and configure an I/O unit. Configuration involves defining the I/O attached to the brain and indicating which features to use, such as scaling for analog points and counters for digital points. Configuration also includes assigning names to your points to identify them.

ioManager and ioControl

You could use ioManager to configure the brain and the I/O modules and points. In fact, ioManager provides specialized configuration options for serial modules, protocols such as PPP, Modbus/TCP, SMTP, and SNMP messaging.

For most necessary configurations, however, ioControl is the easiest and most convenient option for configuration. If you want to learn more about configuring I/O units in ioManager, see the *ioManager User's Guide*, form #1440.

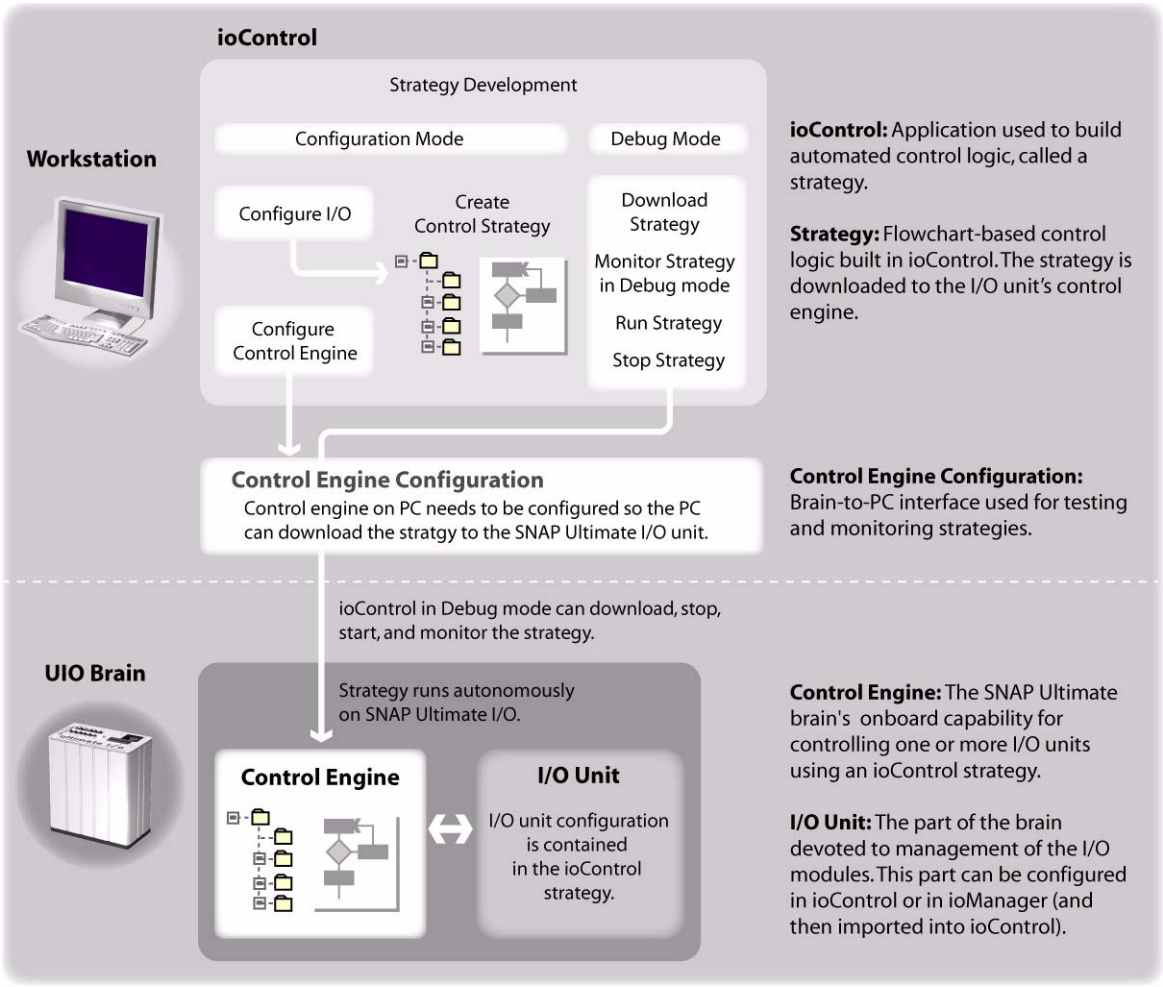


ioControl

In this lesson, you will explore ioControl's application development environment by:

- Creating a new ioControl strategy.
- Adding a control engine so the PC can download the ioControl strategy to the SNAP Ultimate I/O unit.
- Downloading your strategy to the SNAP Ultimate I/O unit.
- Monitoring your points in Debug mode.

You will add control logic in the lessons that follow. The diagram below summarizes the features of ioControl that you will use in this lesson.



ACTIVITY

Creating a New Strategy

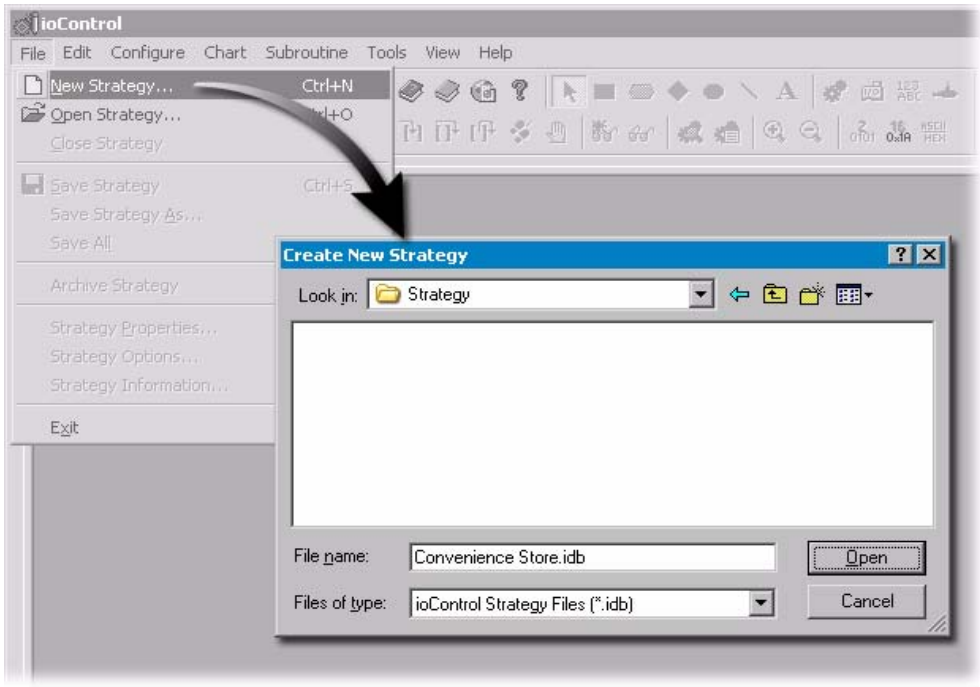
1. Launch ioControl.

From the *Start* menu, select
Programs ⟶ *Opto22* ⟶ *ioProject Software* ⟶ *ioControl* ⟶ *ioControl*.

2. Open a new strategy.

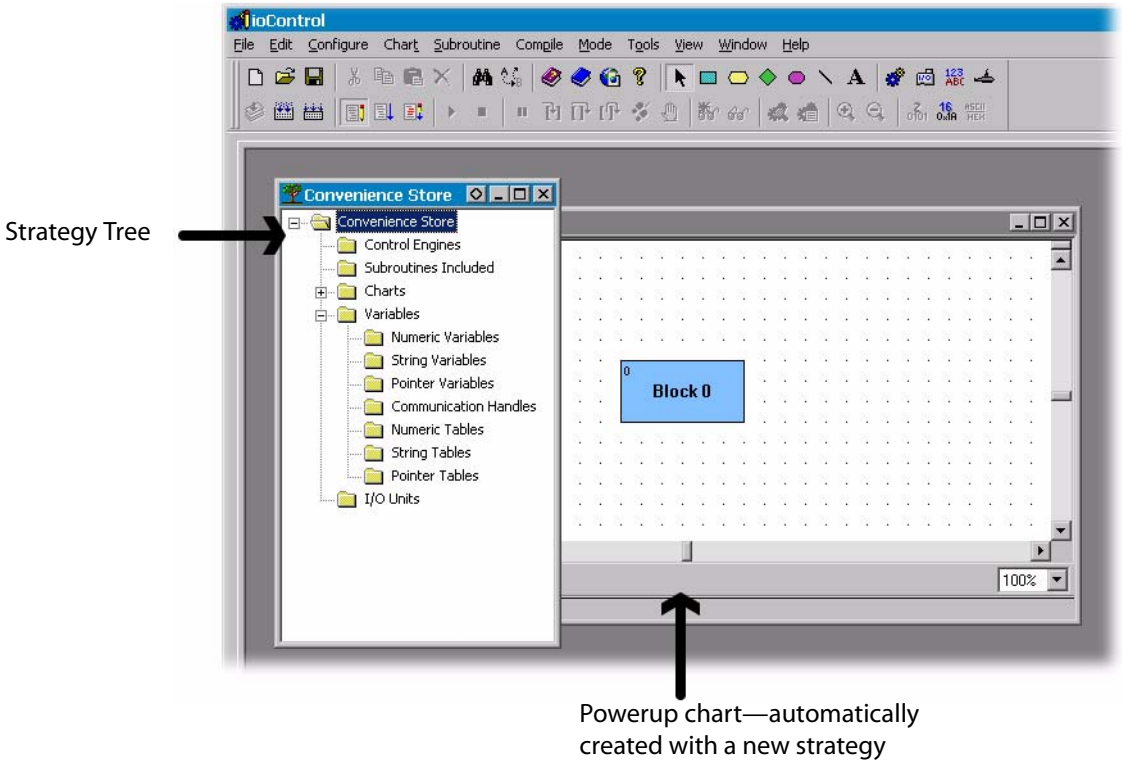
a. To create a new strategy, select *File* ⟶ *New Strategy*.
b. In the Create New Strategy dialog box, navigate to the folder **C:\Program Files\Opto22\UIO Learning Center\Strategy**.

c. In the File name field, type `Convenience Store` as the name of the new strategy.



d. Click *Open*.

This action creates a new strategy, which initially consists of a Strategy Tree and a Powerup chart. The Powerup chart is the starting point for control logic, which you'll add in later lessons.

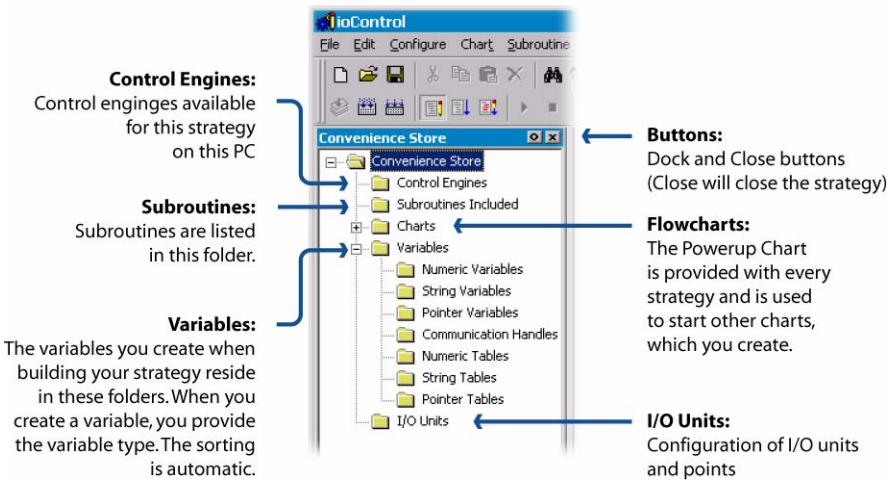


The Strategy Tree presents all the information in your strategy in a simple folder structure. At a glance, you can see the number and types of all I/O points and variables. All charts, configured hardware, and variables are listed for easy reference. Just expand and contract the folders by clicking the plus (⊞) and minus (⊟) symbols to view components. Double-clicking any folder displays dialog boxes for configuring or debugging.

To set up a convenient workspace, you can dock the Strategy Tree.

e. Click the Dock button (◈) on the Strategy Tree.
f. Leave the Strategy Tree in the docked position.

Strategy Tree in docked position:



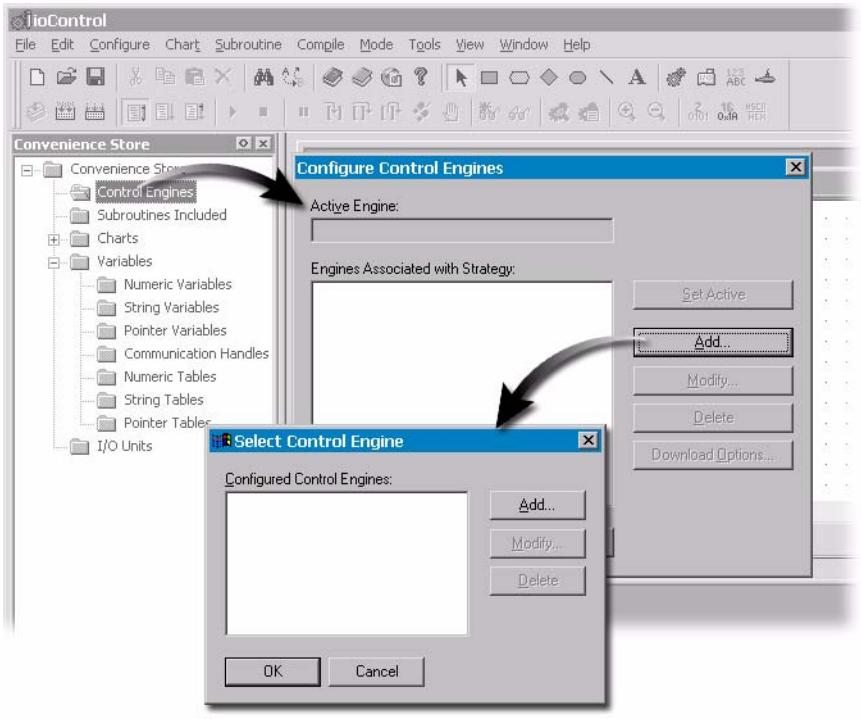
Configuring the Control Engine

First, you will configure the Control Engine, which defines the communication parameters between the host PC and the SNAP Ultimate I/O unit.

1. Open the Control Engine Configuration dialog box.

a. In the Strategy Tree, double-click *Control Engines*.

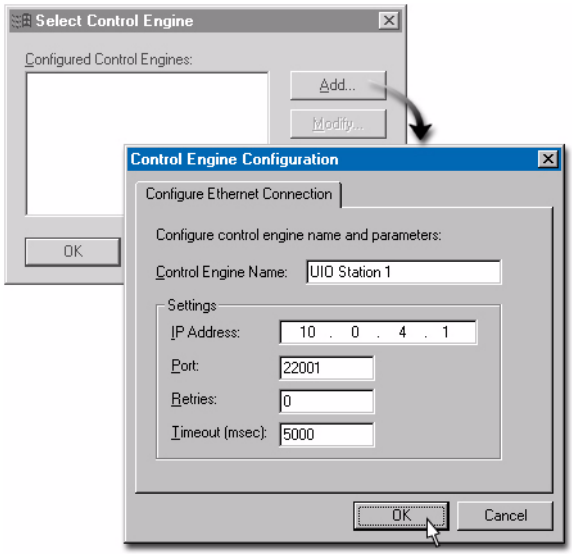
b. In the Configure Control Engines window, click *Add*.



NOTE: If other controllers have been configured on this computer, they are listed in the Select Control Engine dialog box. If this is the first use of ioControl on your workstation, the Configured Control Engines field will be empty.

2. Add Control Engine configuration.

a. In the Select Control Engine window, click *Add* to configure a new control engine.

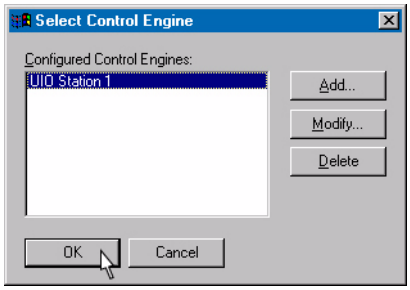


b. Type `UIO Station 1` as the control engine's name.
(Use descriptive names when naming items in ioControl. This practice makes debugging and strategy maintenance easier.)

c. Type the IP Address for your SNAP Ultimate I/O unit.
d. Click *OK* to close the Control Engine Configuration dialog box.

3. Select the active control engine.

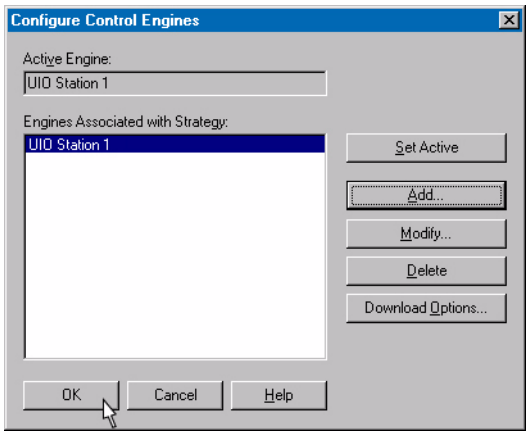
a. Select *UIO Station 1* in the list.
b. Click *OK* to close the Select Control Engine window.



When only one control engine is configured for use with a strategy, that control engine is automatically designated the active control engine. The active control engine is the one to which the strategy will be downloaded. You can configure several control engines, but only one is active at any time.

If more than one control engine appears in the Configure Control Engines window and the control engine you wish to use does not appear in the Active Engine field, select the control engine from the list and click the *Set Active* button. The selected control engine name is then shown in the Active Engine field.

Notice that control engines can be added, deleted, and modified from this window.

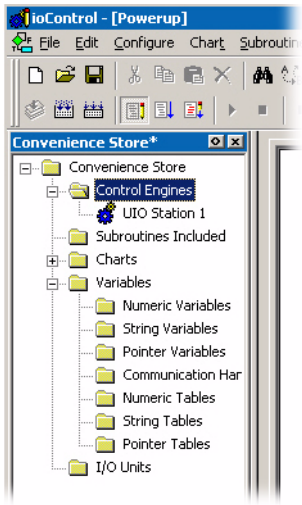


c. Click *OK* to close the Configure Control Engine window.

d. Click the plus sign (⊞) to the left of *Control Engines* in the Strategy Tree to expand the folder.

   Notice that your control engine is now listed in the Strategy Tree.

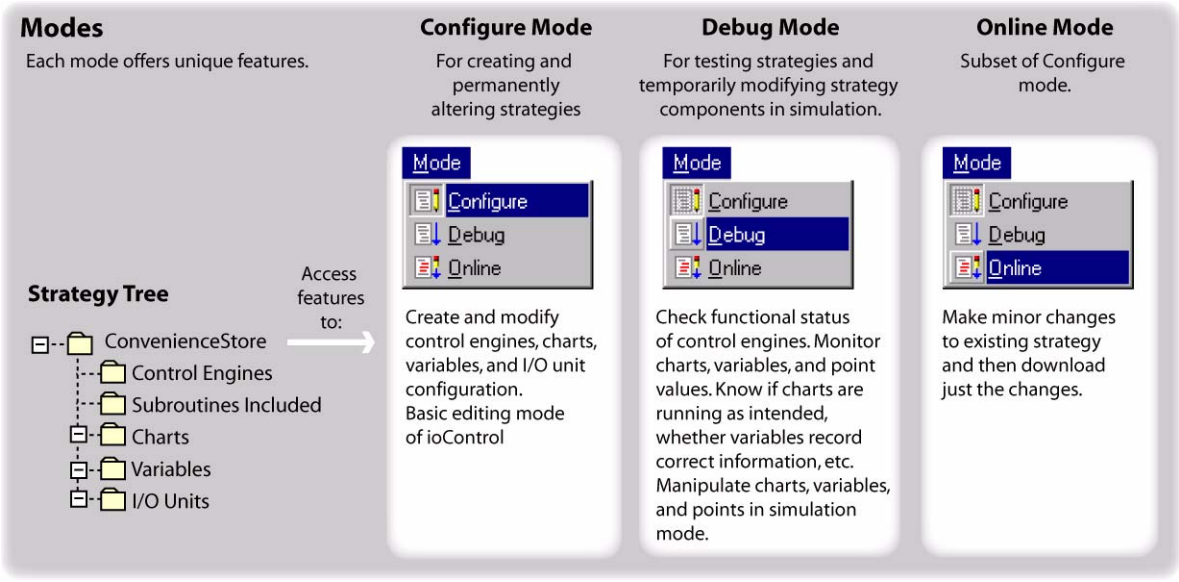
You've just used ioControl in Configure mode. To test your communications with the control engine, you will use ioControl's Debug mode.



About Configure, Debug, and Online Modes:

An ioControl strategy is written and edited in Configure mode. When you switch to Debug mode, ioControl saves, compiles, and downloads the strategy to the control engine. In Debug mode, you can examine and modify I/O point states and variable values. You also can monitor the step-by-step execution of your ioControl strategy.

TIP: If you can't locate a feature or dialog box, check that you are in the correct mode.



4. Verify control engine communications.

To test communications, you can download your strategy to the control engine. At this point, your strategy is essentially empty, but this step will ensure that control engine communications are correct.

a. Select *Debug* from the *Mode* menu. (You can also click the Debug icon in the toolbar.)

This action saves, compiles, and downloads the strategy.

Switching to Debug mode prompts you with messages:

b. Save your strategy when prompted.

If this is the first strategy to be downloaded since power has been cycled to the control engine, an ioControl window will open with a "Powerup Clear Expected returned from control engine" message.

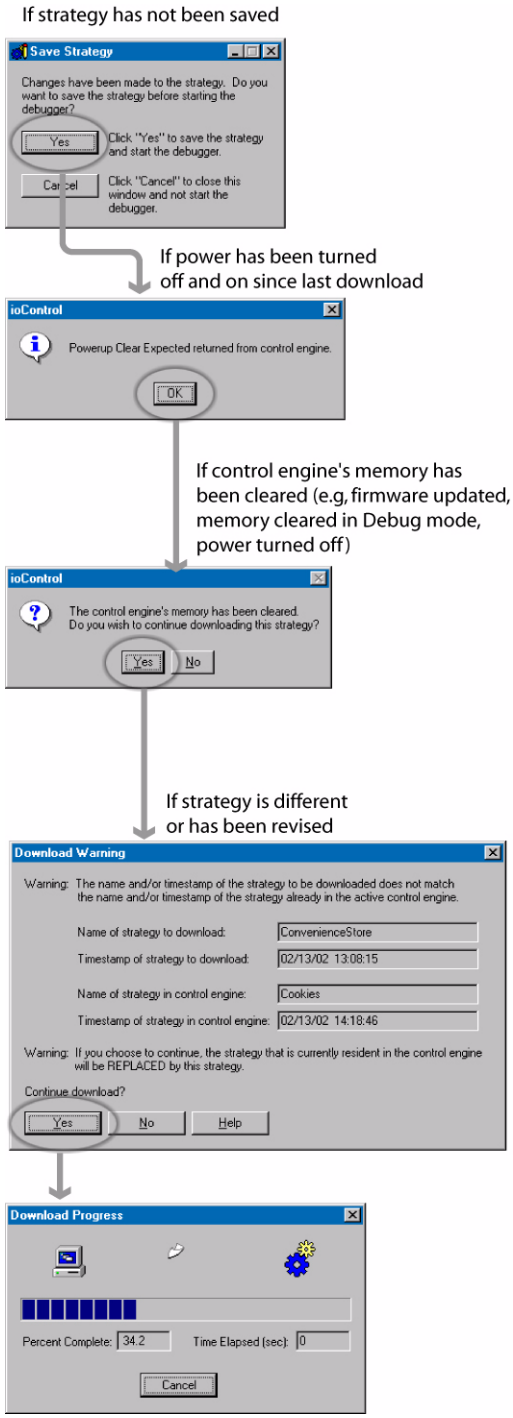
c. Acknowledge the message by selecting *OK*.

Occasionally the control engine memory may be cleared. This can occur if the firmware file was just downloaded, if the memory was cleared in Debug mode, or if the power was turned off and the battery was dead.

d. Click *Yes*.

If the strategy running on the control engine is different from the strategy being downloaded, a Download Warning window opens.

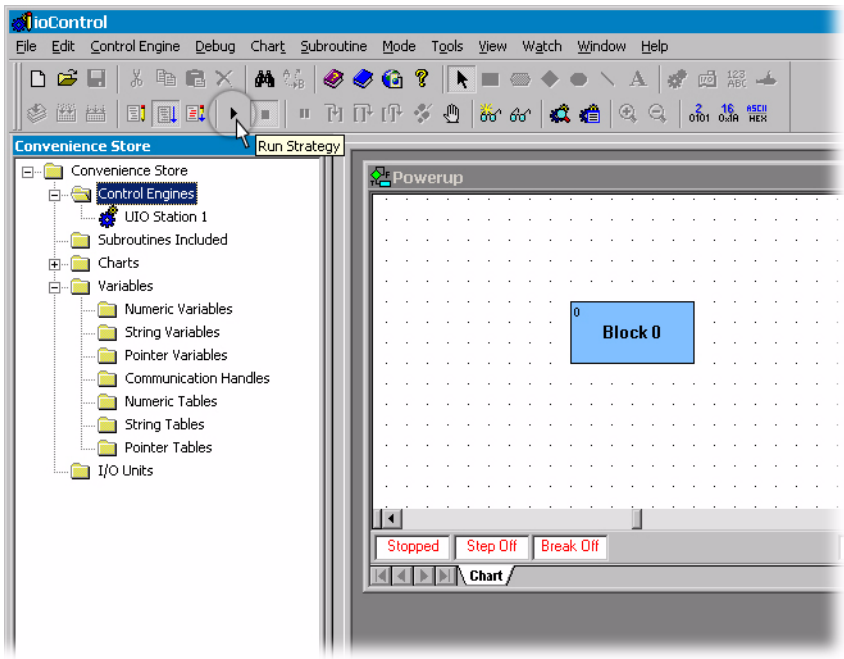
e. Click *Yes*.



NOTE: If you encounter problems downloading or running your strategy, see the Troubleshooting appendix in the *ioControl User's Guide* (Opto 22 form 1300). User documentation is on the SNAP Ultimate I/O Learning Center CD in the Manuals folder.

5. Run the strategy.

a. Click the *Run* button ( ) to start the strategy.



b. Look at the lower right-hand corner of the ioControl window. If there are no errors, you are communicating successfully.

ioControl Menu and Toolbars in Debug Mode:

When you switched to Debug mode, new tools became available to you:

ioControl Menu and Toolbars in Debug Mode

Configure, Debug, and Online mode buttons. The selected button (Debug) indicates the current mode.

Run, Stop, and Pause buttons

Debugging tools

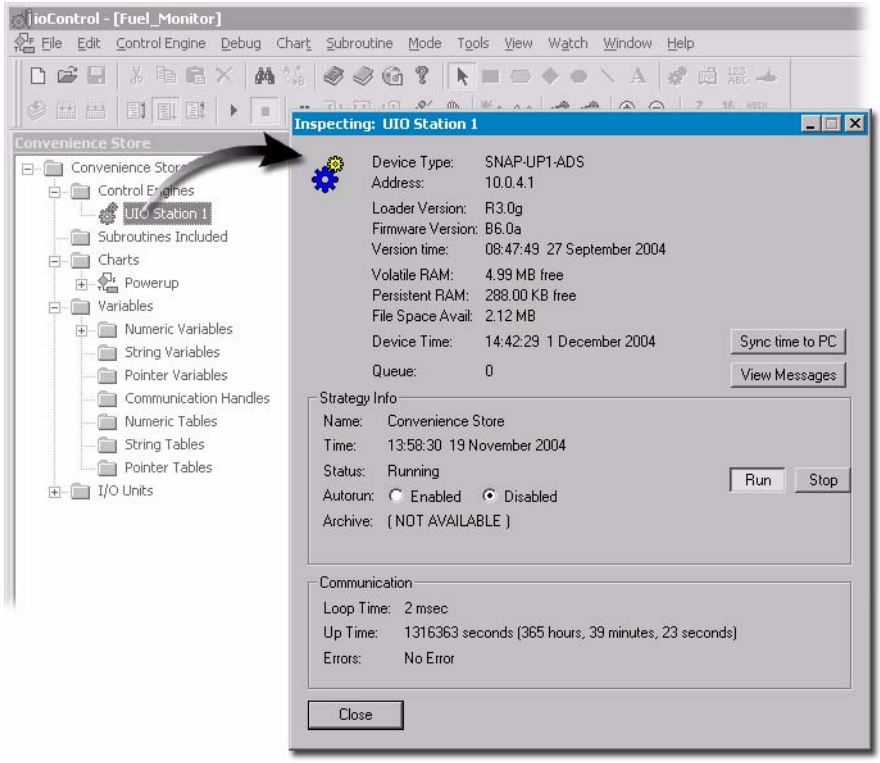
Watch window buttons

Inspect control engine

Sniff communications

6. View control engine status.

Double-click your control engine, *UIO Station 1*, in the Strategy Tree.



The Inspecting dialog box describes the versions of the loader and firmware on the brain. It also provides information about the brain's memory:

- A SNAP Ultimate brain has 8 MB of RAM on the control side, 256K of which is battery-backed.
- Volatile RAM shows the amount of total RAM available for use.
- Persistent RAM shows the amount available in battery-backed RAM, where persistent variables, variables initialized on download, the autorun flag, and the strategy archive are stored.

Loop Time is the time required to gather the inspection data from the control engine (time taken for a single transaction). For more information on the Inspecting dialog box, see the ioControl User's Guide, Opto 22 form 1300.

Autorun

Autorun determines whether the strategy starts automatically when downloaded or when the control engine is restarted.

Users of older Opto 22 controllers will notice that the autorun flag in the SNAP Ultimate I/O control engine is a software setting, instead of a hardware jumper setting.

In ioControl, the autorun flag can be set in two places in the strategy download options:

- In the Inspecting Controllers dialog box.
- In the Download Options found under *File* ⟶ *Strategy Options*, *Download* tab.

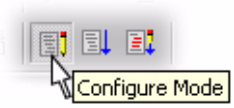
Archive

The archive feature says "Not Available" because an archive of the strategy has not been downloaded to the brain. In Configure mode, you can set the strategy options to also archive the strategy to the brain upon download. See *File* ⟶ *Strategy Options*, *Archive* tab.

7. Return to Configure mode.

At this point the strategy is still running. You can leave it running as you switch back to Configure mode.

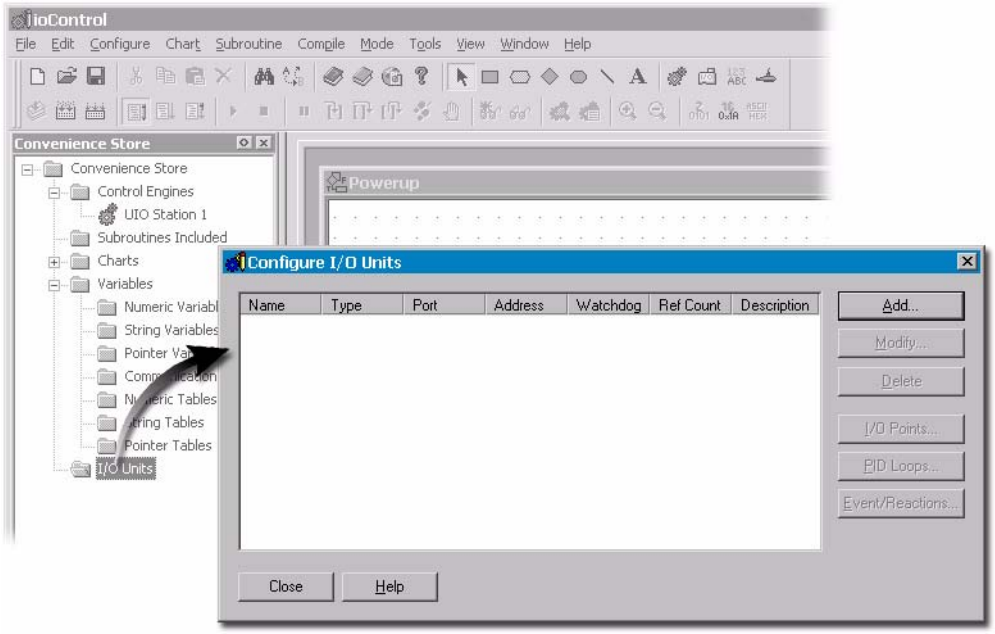
a. Close the Inspecting dialog box.
b. Click the Configure mode button on the toolbar or choose *Mode* ➞ *Configure*.



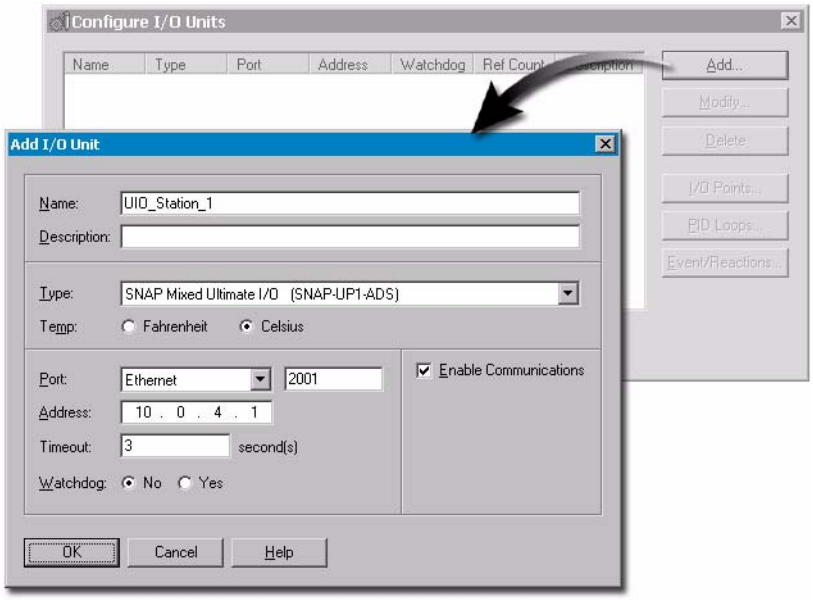
Configuring the I/O Unit

1. Configure the I/O Unit.

a. Double-click *I/O Units*. You can also right-click I/O Units and choose *Configure*.



b. Click the *Add* button or double-click in the list box to open the Add I/O Unit dialog box.



c. Type `UIO Station 1` in the Name field. Notice that spaces are converted to underscores.

d. Select *SNAP Mixed Ultimate I/O (SNAP-UP1-ADS)* as the Type.

e. Select *Fahrenheit*.

f. Type the IP address you've designated for the I/O unit.

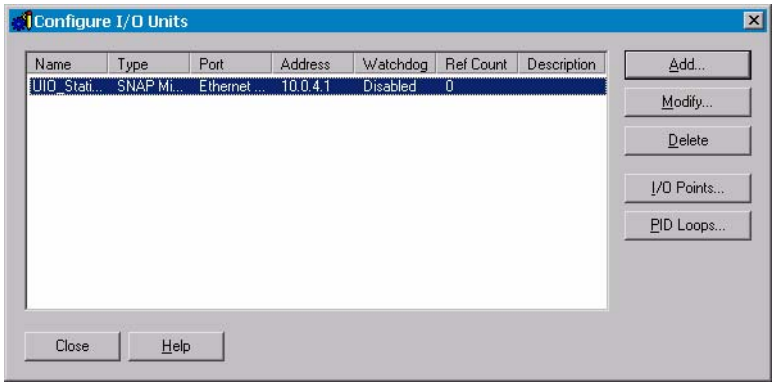
   There is no need to change the default Port or Watchdog settings.

g. Click *OK* to close the Add I/O Unit dialog box.

h. Leave the Configure I/O Units dialog box open for the next steps.

About Watchdog

The Watchdog option is used to monitor activity on the port. When a watchdog is enabled and no communication is received for the specified time interval, the unit assumes a watchdog state, which sets designated digital and analog output points to pre-determined values. In this strategy we are not using watchdogs.

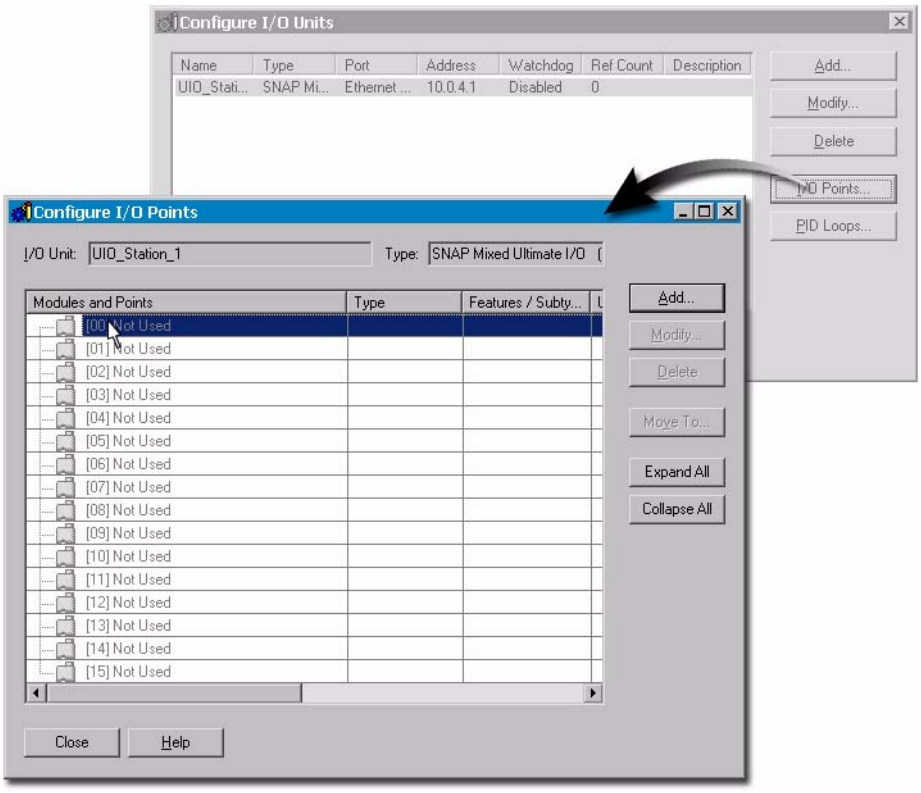


Configuring I/O Points

Configuration involves identifying each module attached to the brain and specifying which points are in use. (Configuration prepares the I/O unit for use by ioControl flowcharts, which will be explained in Lesson 3.) I/O points can be configured on-the-fly (as you need them when creating flowcharts) or all at once (before they are used). In this exercise, you will configure the I/O points that you will need using the all-at-once method.

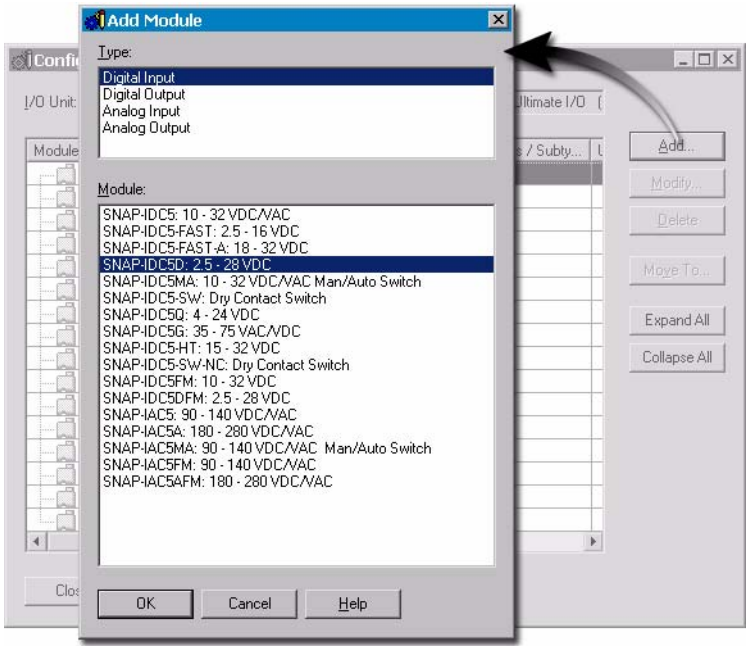
1. Define a digital input module.

a. Click the *I/O Points* button.



b. Select module *[00]*.

c. Click the *Add* button to configure this module.



d. Select *Digital Input* as the Type.
e. Select *SNAP-IDC5D: 2.5 - 28 VDC/VAC* as the Module.
f. Click *OK* to close the Add Module dialog box.

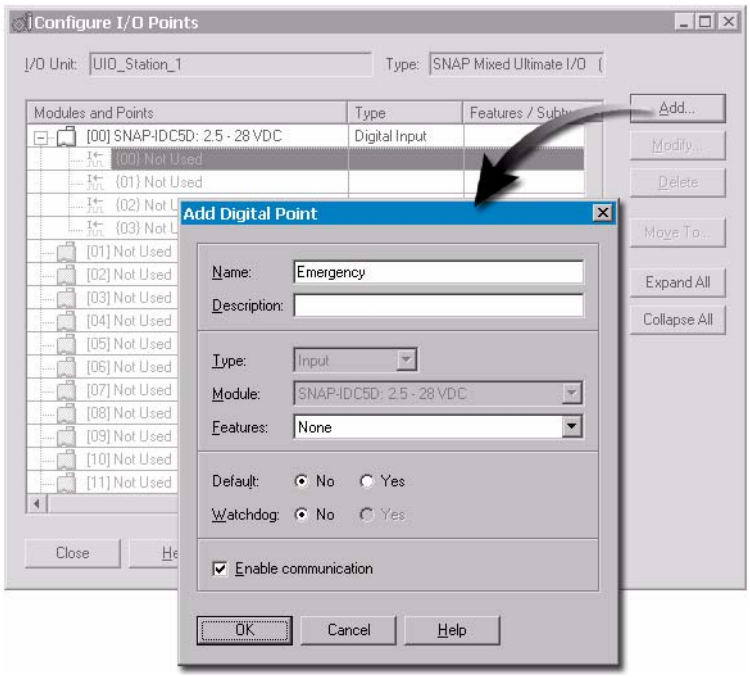
The module is now defined in the Configure I/O Points dialog box.

2. Configure a digital input point.

a. Click the plus symbol (⊞) next to the digital input module you just added.

Now you can see the available points. As this is the first module on the rack, its points are numbered 00, 01, 02 and 03.

b. Double-click Point *[00]*, or select Point *[00]* and click the *Add* button.



c. Type `Emergency` for the point name.
d. Click *OK* to close the Add Digital Point dialog box.

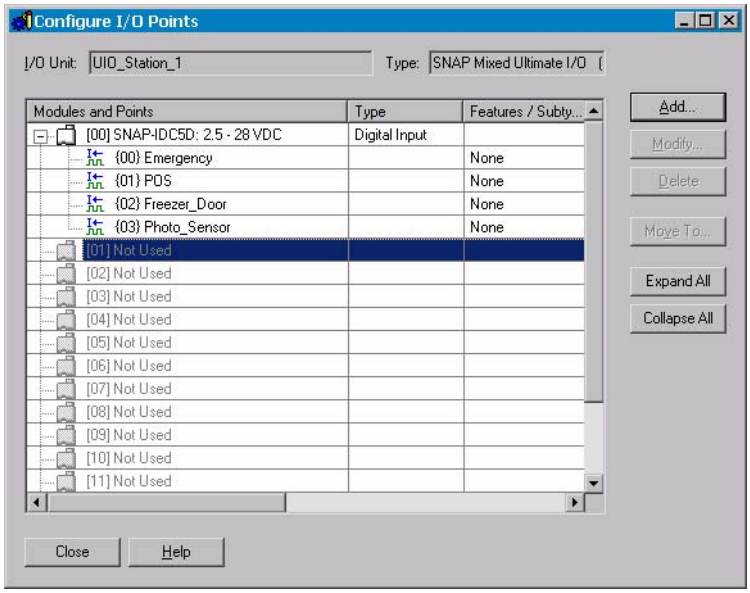
3. Configure the remaining digital points of module 0.

a. Repeat Step 2 for the remaining points on module 0.
b. Double-click each point to access the Add Digital Point dialog box and assign it a point name. Use the following names for points 01, 02, and 03:

Point	Name
01	POS
02	Freezer_Door
03	Photo_Sensor

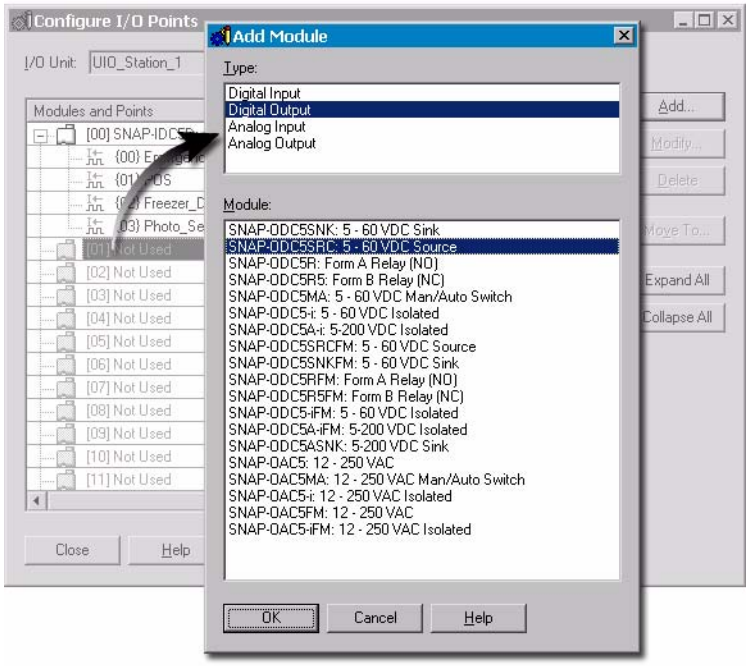
NOTE: Type the point names exactly as shown here.

When complete, your Configure I/O Points dialog box will appear as follows:



4. Configure a digital output module.

a. Double-click Module *01.*
b. Choose *Digital Output* for Type.
c. Choose *SNAP-ODC5SRC: 5 - 60 VDC Source* for Module.

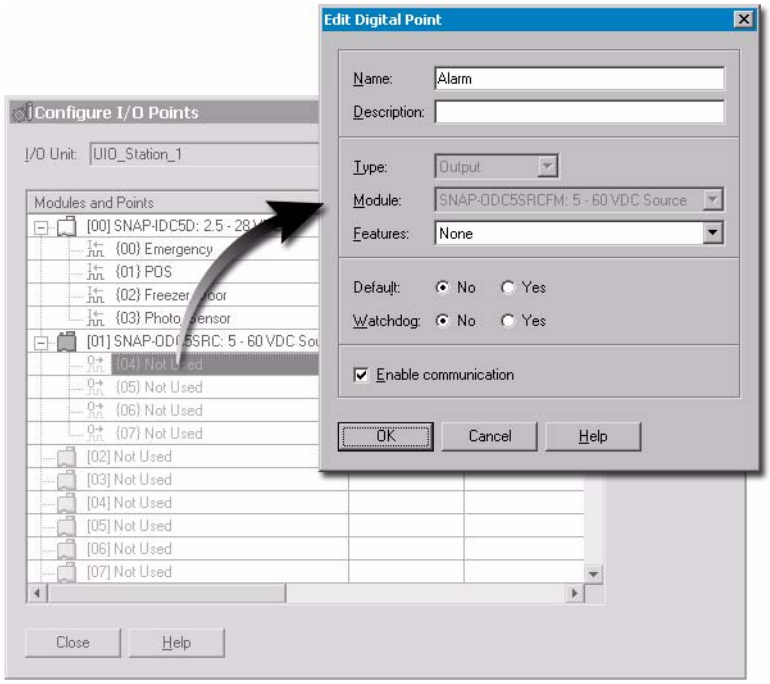


d. Click *OK* to close the Add Module dialog box.
e. Expand the points for Module 01 (click ⊞).

5. Configure digital output points.

Configure your Digital Output module and points with the information shown below, as described in the following instructions.

a. Double-click *Point 04*.



b. Type `Alarm`.

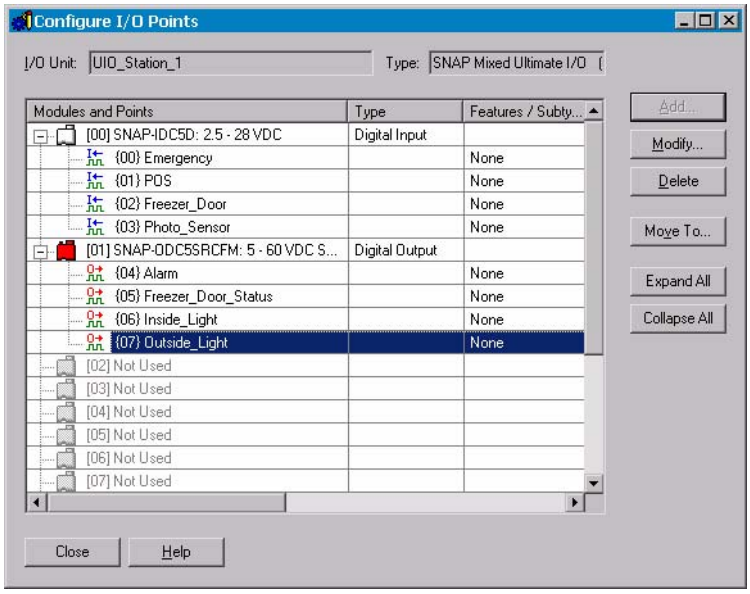
c. Click *OK*.

d. Repeat Steps 5a through 5c for points 05, 06, and 07, assigning the names in the following table.

I/O Configuration—Points 4–7

Module	I/O Type	Module Type	Point	Name in Course	Scaling
1	Digital Output	SNAP-ODC5SRC: 5 - 60 VDC Source			
			4	Alarm	
			5	Freezer_Door_Status	
			6	Inside_Light	
			7	Outside_Light	

When you are finished, your Configure I/O Points dialog box should appear as follows:

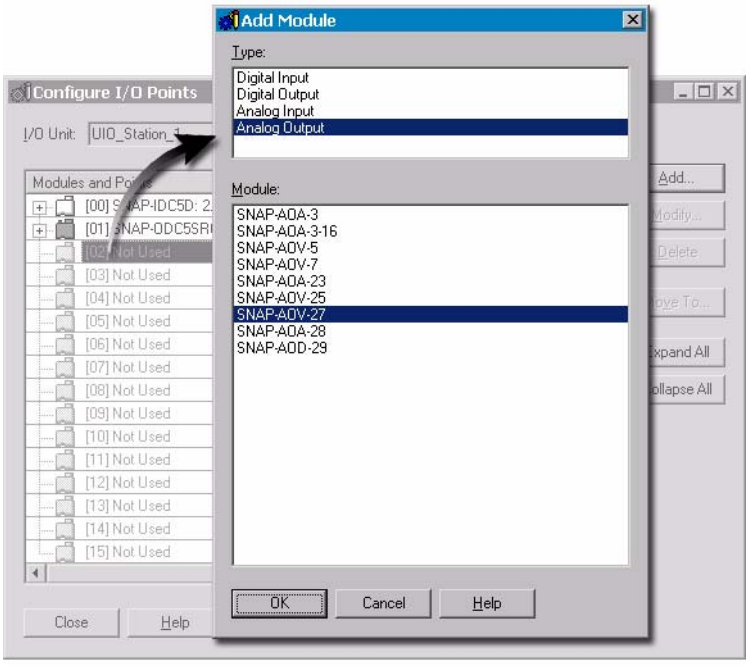


6. Collapse the points from modules 00 and 01.

Click the minus symbols (⊟) next to Modules 00 and 01.

7. Configure an analog module.

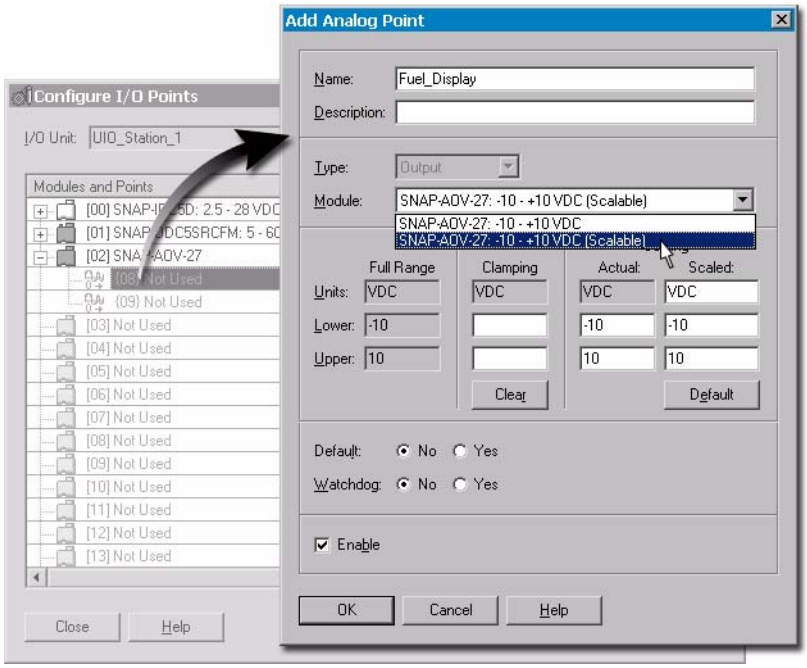
a. Double-click Module *02* (or select Module *02* and click *Add*).



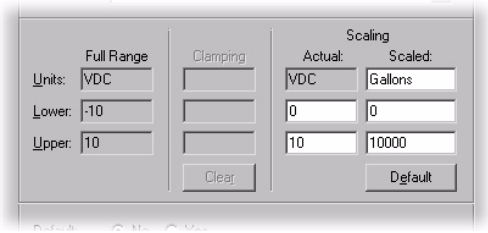
b. Select *Analog Output* as the Type.
c. Select *SNAP-AOV-27* as the Module.
d. Click *OK* to close the Add Module dialog box.

8. Configure and scale an analog point.

a. Expand the points below Module 02.
b. Double-click *Point 08*.
c. Type `Fuel_Display` for the Name.
d. Choose *SNAP-AOV-27: -10 - +10 VDC (Scalable)* for Module.



e. Type the following values in the Scaling group:



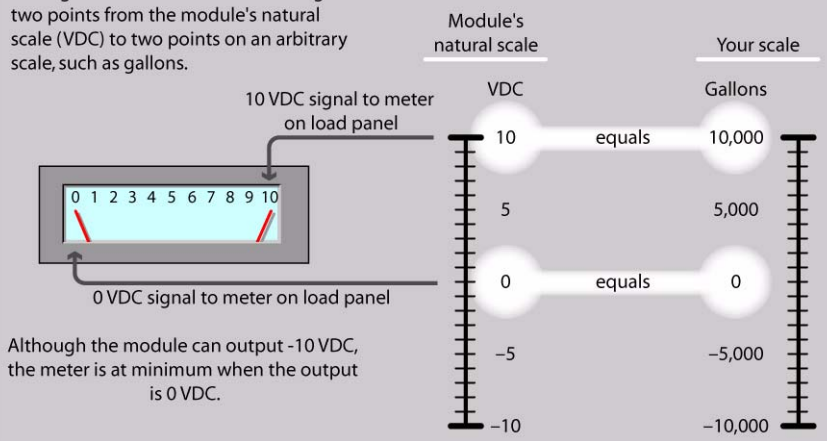
- Units: `Gallons`
- Lower – Actual: `0`
- Lower – Scaled: `0`
- Upper – Actual: `10`
- Upper – Scaled: `10000` (NOTE: Do not use a comma separator, and make sure to type 10000 instead of 1000.)

f. Click *OK* to close the Add Analog Point dialog box.

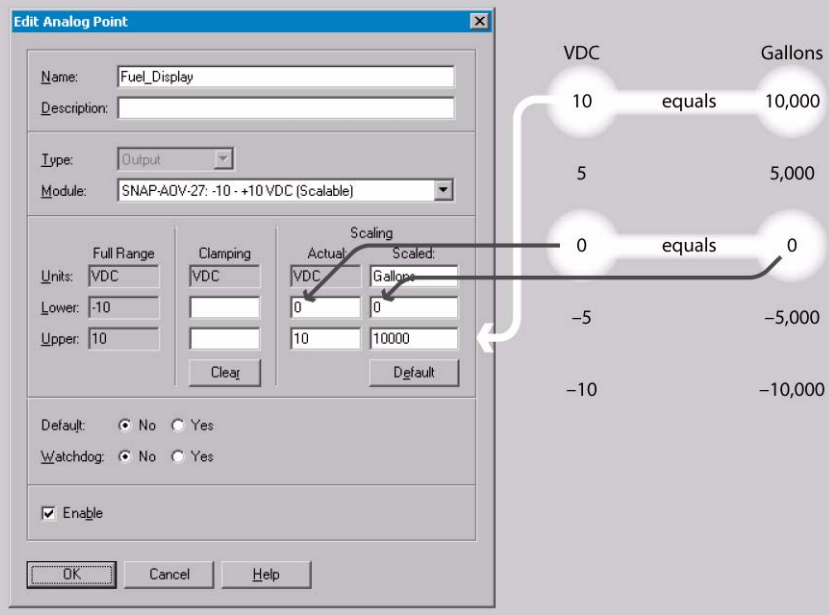
About Scaling

Scaling your meter allows you to correlate the natural scale of the module, which is in voltage, to a more understandable value, such as gallons.

Scaling your meter



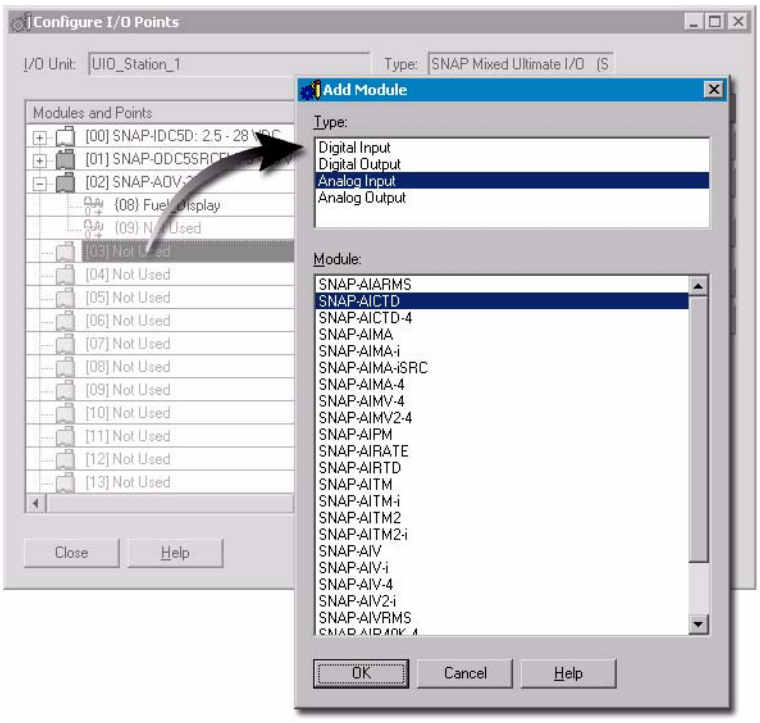
Scaling the Meter: Inserting the Correct Values



9. Configure module 3.

a. Double-click module *03* (or select module *03* and click *Add*).
b. Select *Analog Input* as the Type.

c. Select *SNAP-AICTD* as the Module.

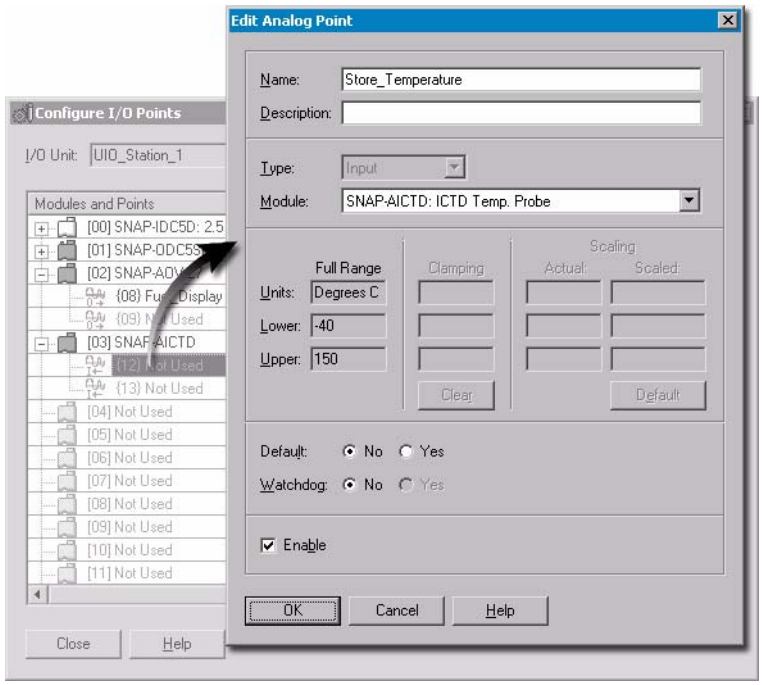


d. Click *OK* to close the Add Module dialog box.

10. Configure point 12.

a. Expand the points below module 03.
b. Double-click point *12.*

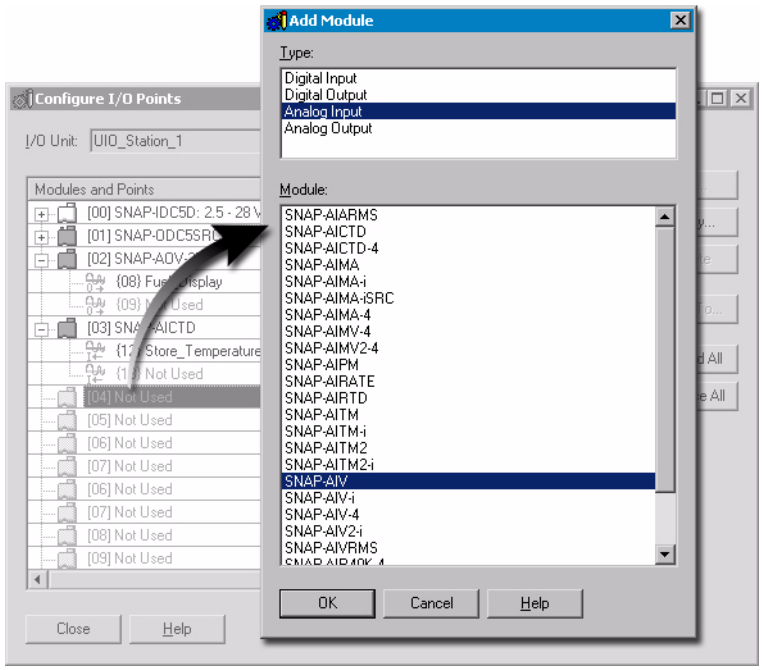
c. Type `Store_Temperature` for the Name.



d. Click *OK* to close the Add Analog Point dialog box.

11. Configure module 4.

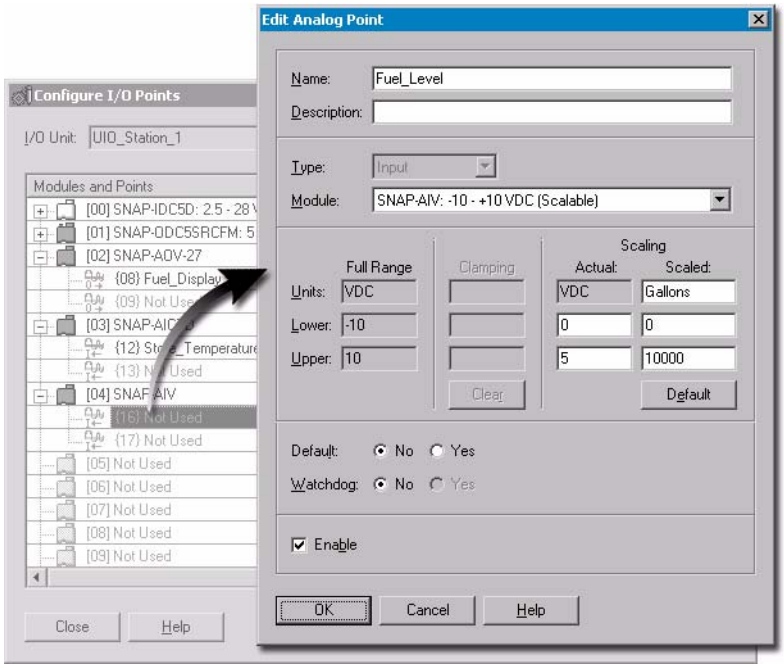
a. Double-click Module *04* (or select Module *04* and click *Add*).
b. Select *Analog Input* as the Type.
c. Select *SNAP-AIV* as the Module.



d. Click *OK* to close the Add Module dialog box.

12. Configure Point 16.

a. Expand the points below module 04 and double-click Point *16*.
b. Type `Fuel_Level` for the Name.
c. Choose *SNAP-AIV: -10 - + 10 VDC (Scalable)* for Module.



d. Type the following values in the Scaling group:
   - Units: `Gallons`
   - Lower – Actual: `0`
   - Lower – Scaled: `0`
   - Upper – Actual: `5`
   - Upper – Scaled: `10000`

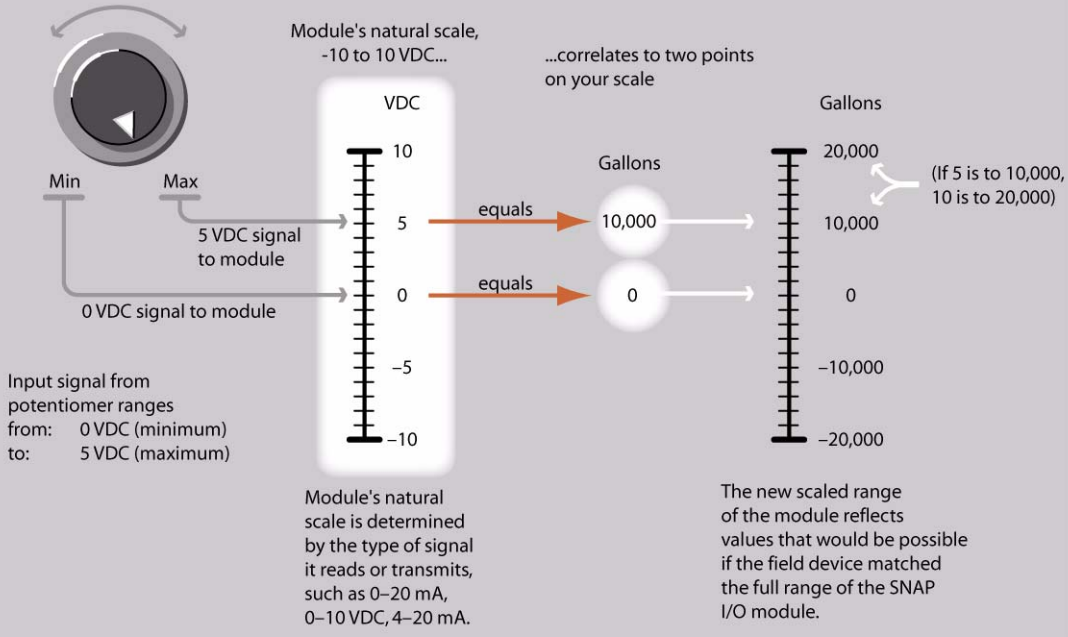
e. Click *OK* to close the Add Analog Point dialog box.

Scaling the Potentiometer

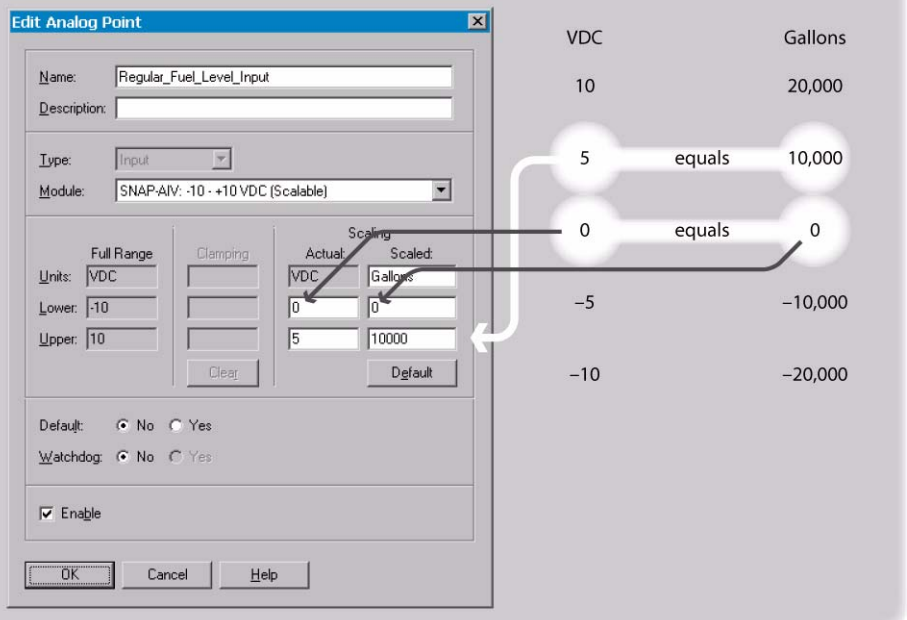
The range of the potentiometer is different from the meter's range, but since we need the two to correlate, we used different settings so both devices will use the same range.

Scaling Analog Points

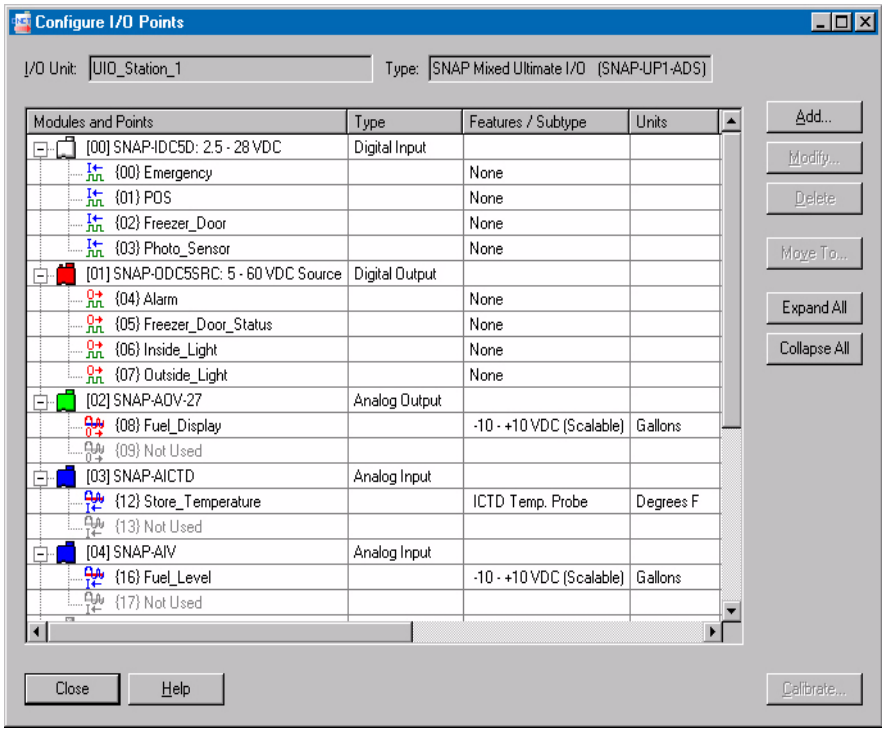
Scaling this potentiometer involves correlating two points from the module's natural scale (VDC) to two points on an arbitrary scale, in this case, gallons.



Scaling the Potentiometer: Inserting the Correct Values



The Configure I/O Points dialog box no shows this configuration information:



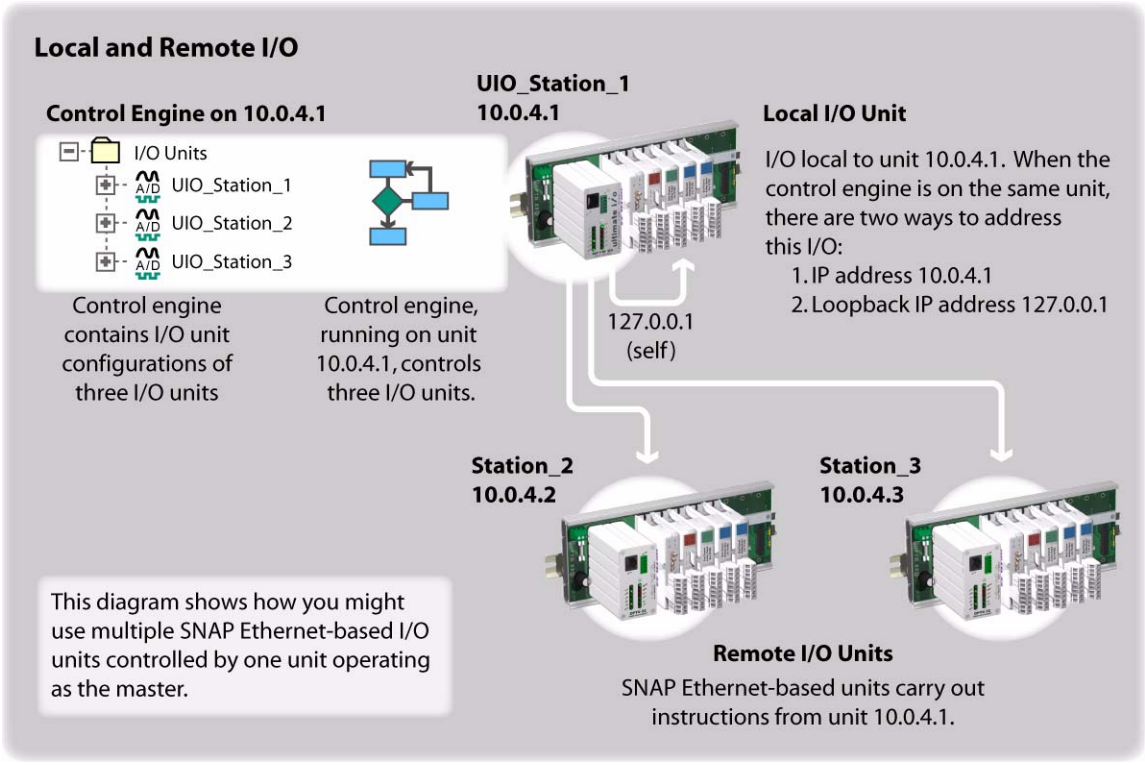
I/O Configuration

Module	I/O Type	Module Type	Point	Name in Course	Scaling
0	Digital Input	SNAP-IDC5D: 2.5-28 VDC/VAC			
			0	Emergency	
			1	POS	
			2	Freezer_Door	
			3	Photo_Sensor	
1	Digital Output	SNAP-ODC5SRC: 5 - 60 VDC Source			
			4	Alarm	
			5	Freezer_Door_Status	
			6	Inside_Light	
			7	Outside_Light	
2	Analog Output	SNAP-AOV-27			
		-10– +10 VDC (Scalable)	8	Fuel_Display	-10000/10000 gal.
3	Analog Input	SNAP-AICTD			
			12	Store_Temperature	
4	Analog Input	SNAP-AIV			
		-10– +10 VDC (Scalable)	16	Fuel_Level	-20000/20000 gal.

13. **Close the Configure I/O Points dialog box.**

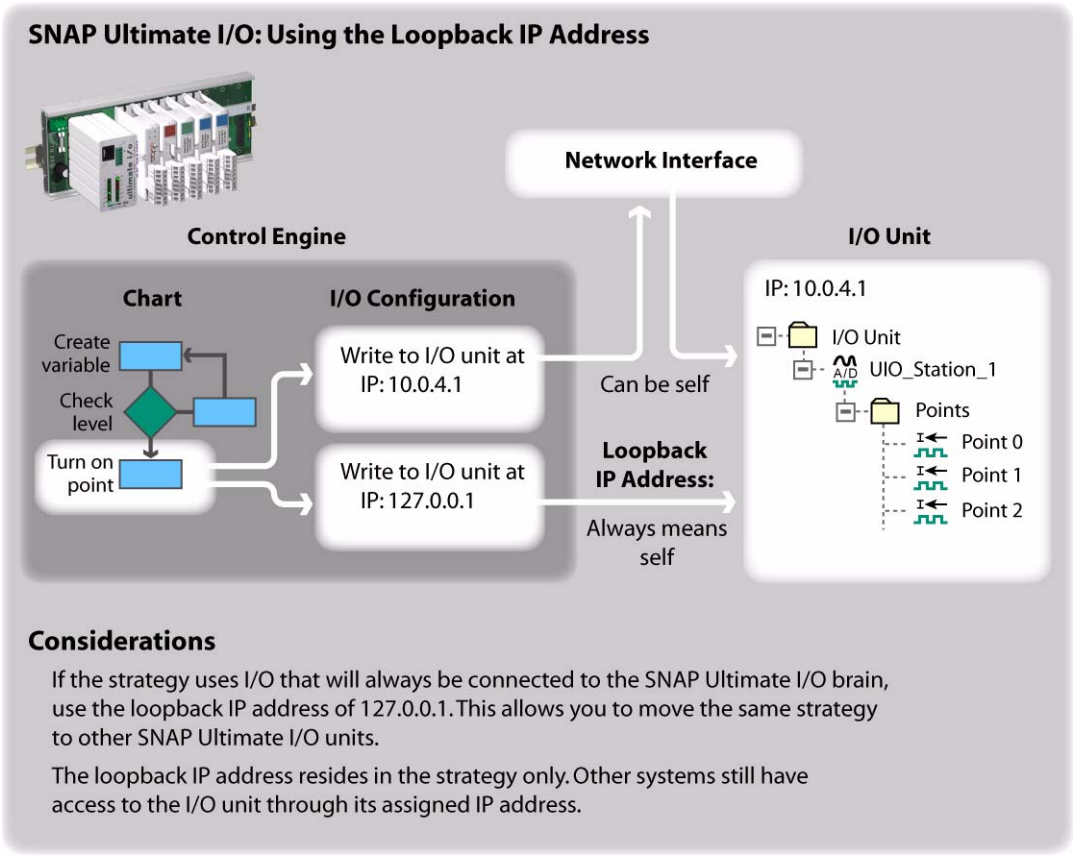
Making the Strategy Independent of Specific Controllers

If you have more than one SNAP Ethernet-based I/O unit, you could build a system in which the SNAP Ultimate I/O unit controls all the units with one strategy. This diagram shows a SNAP Ultimate I/O unit controlling other units. Notice that the I/O unit running the control strategy has local I/O, meaning the I/O is on the same unit as the control engine.



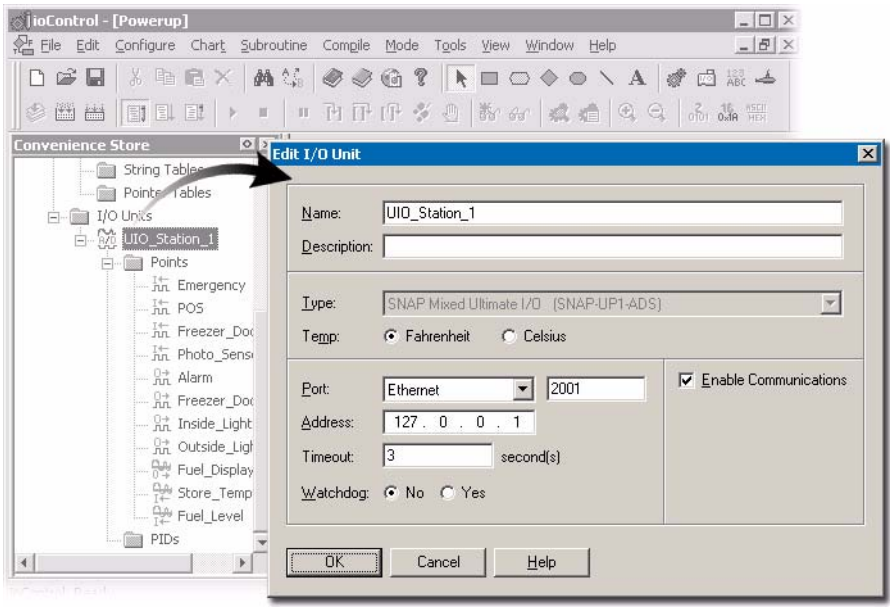
As you build your strategy, you will add commands that read from and write to points. The strategy will find these points using the IP Address defined in the I/O Unit properties. By using the brain's IP address within the I/O unit configuration (e.g., 10.0.4.1), you hard-coded all interaction with the I/O at the named IP Address. Next,

you'll see how to configure your I/O unit so that it is always local regardless of which IP address you may assign to the SNAP Ultimate I/O unit.



1. Open the Edit I/O Unit dialog box.

a. Double-click *UIO_Station_1* in the Strategy Tree.



b. Type `127.0.0.1` for the IP Address.
c. Click *OK* to close the Edit I/O Unit dialog box.

2. Download and Run your Strategy.

a. Choose *Mode* ⟶ *Debug* or click the Debug button on the toolbar.
b. Acknowledge any download messages.
c. Choose *Debug* ⟶ *Run*.

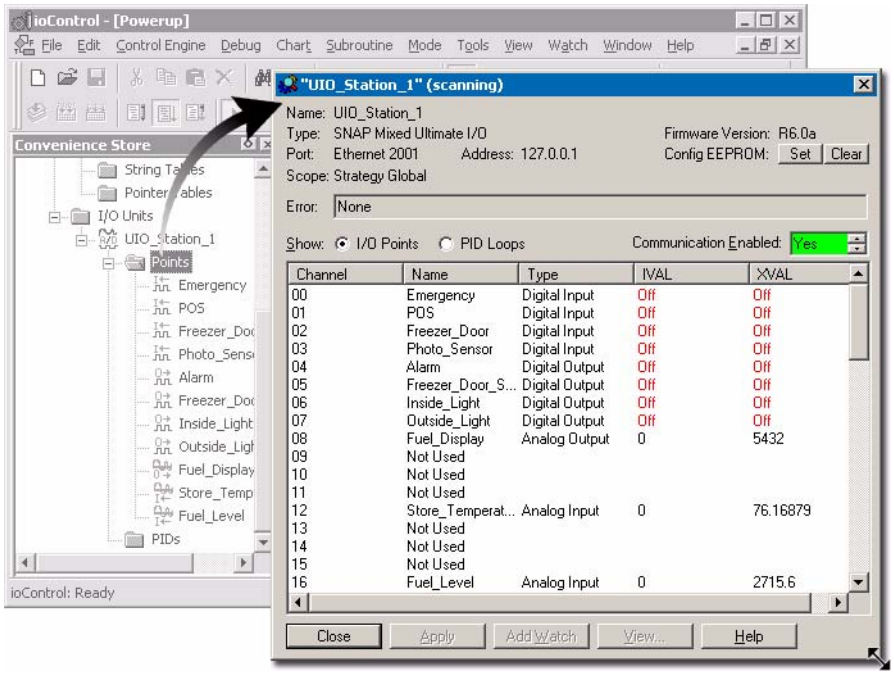
Observing Points in Debug Mode

Now that your I/O points are configured, you can observe the status of the points while in ioControl's Debug mode.

1. Open the View I/O Unit dialog box.

a. Double-click the *Points* folder.

b. Expand the Points dialog box.



You can use this dialog box while your strategy is running to monitor points and to force input or output values.

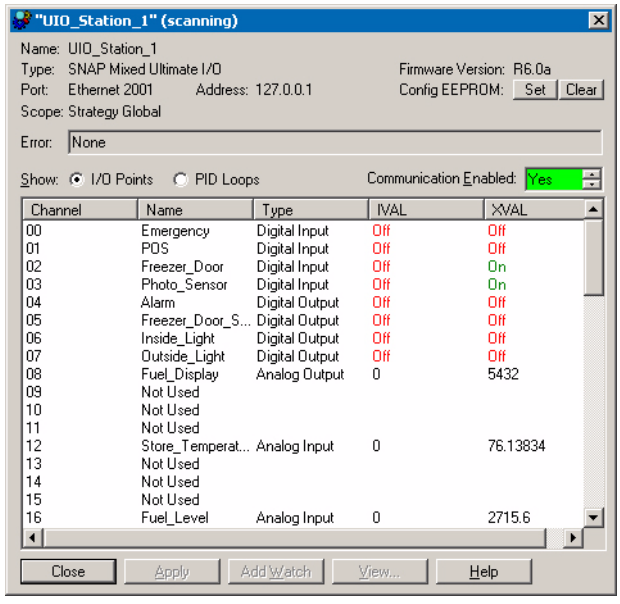
2. Observe point states and values.

a. While viewing the I/O unit, turn on and off the switches on your Learning Center.

You will notice the states of these points in the XVAL column.

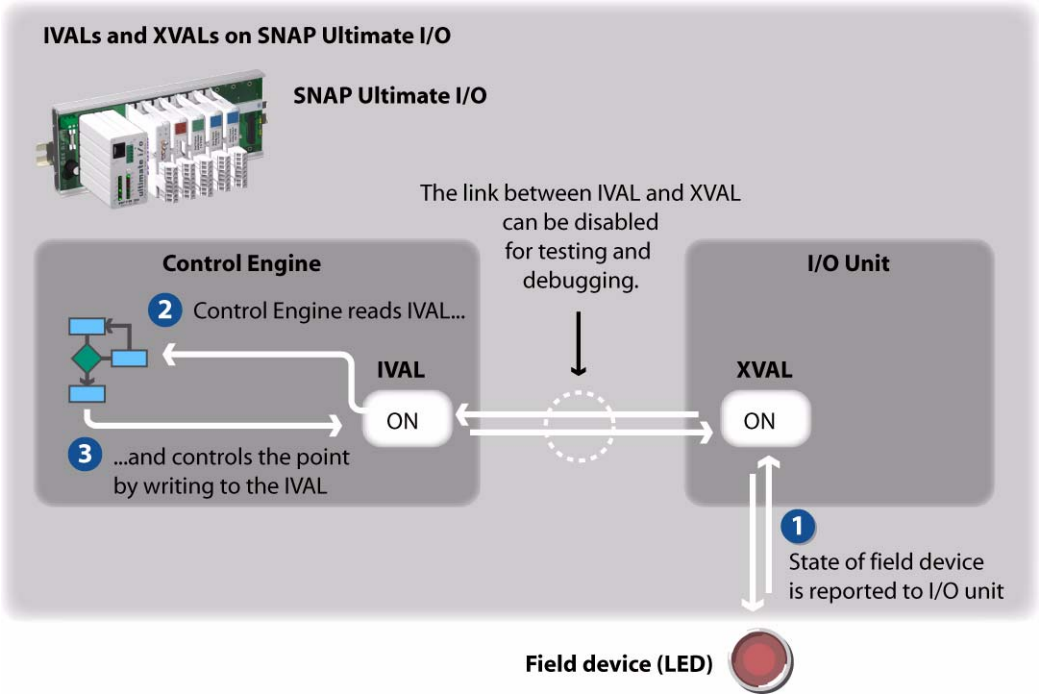
b. Turn your Fuel Level potentiometer and notice its value in the XVAL column.

NOTE: You may need to scroll down the list to see channel 16.



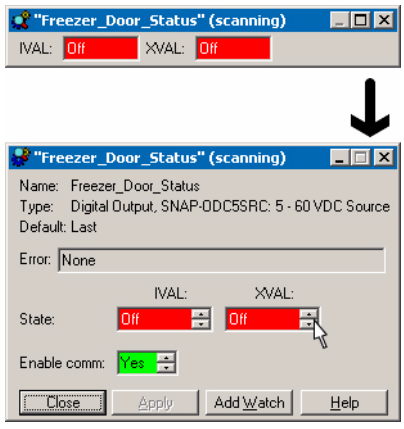
Writing to Outputs and Simulating Inputs in Debug Mode

ioControl has two representations of a point's state or value. One representation is the true state being reported by the I/O module, which is called the XVAL (X for external). The IVAL (I for internal) represents the value within ioControl of any point that is referenced in the strategy. (Points not used in the strategy will not have updated IVALs.) The internal and external values are linked, meaning their values or states are identical. However, when debugging your strategy, you may wish to disable the link between the IVAL and XVAL to test how your ioControl strategy would react to hypothetical states and values. For example, to test your strategy with an extreme condition, you may need to disable an input to simulate an out-of-tolerance value.



1. Write to an output point.

a. In the View I/O Unit dialog box, double-click *Point 05, Freezer_Door_Status*.
b. Click the *Maximize* button ( ).



You can write to an output point by changing the XVAL:

c. Click the up-down arrows in the XVAL field to change the Off state to On.
d. Click *Apply*.

Notice that your Freezer Door Status LED is on.

e. Repeat to turn off the light.
f. Click *Close*.

2. Examine other points.

You can open other points from the View I/O Unit dialog box (NOTE: Point 04 controls an audible alarm, which you may not wish to turn on until you're confident that you can promptly turn it off):

a. Double-click any of the points in the View I/O Unit dialog box.
b. Expand their view dialog boxes.
c. Turn on and off the IVALs and XVALs and click *Apply*.

NOTE: If you want to write to an output point, you have to use the XVAL. If you try to change the XVAL of an input point, it will be overwritten by the true setting of the field device.

3. Return to Configure mode.

a. When finished examining points, close all dialog boxes.
b. You may leave your strategy running.
c. Choose menu command *Mode ⟶ Configure*.

You will continue using ioControl in the next lesson.

ioControl: Flowcharts

Skills

ioControl

- Creating a Flowchart
- Choosing Flowchart Colors and Text
- Closing Your Chart
- Adding a Command to the Powerup Chart
- Using the Flowchart Drawing Tools
- Adding an Instruction
- Completing an Instruction
- Archiving Your Strategy
- Setting Workspace Options

Scenario

In the previous lesson, you created a strategy using ioControl. The strategy includes the control engine you defined and the I/O you configured. The I/O configuration allows you to actuate any devices and read any sensors connected to your I/O modules. This level of control allows you to test and modify your system from a remote location, but it lacks the control logic needed to automatically turn on and off lights, trigger alarms, and record activities and transactions. To have these capabilities, you need programming logic, which you create using flowcharts in ioControl.

In this scenario, your ioControl strategy controls one SNAP Ultimate I/O unit; however, this isn't the only possible use of these systems. For example, you may have several SNAP Ultimate I/O systems, each controlling a different process and therefore running different strategies. Or you may have two or more SNAP Ultimate I/O systems controlling identical processes in different areas and running the same strategy. You can also deploy one SNAP Ultimate I/O unit with a strategy and other SNAP Ethernet-based I/O units with no strategy that are controlled by the master I/O unit.

You are beginning the major application development phase of your project. Over the next few lessons you will add programming logic and variables to your strategy while frequently testing your changes.

Concepts

Strategy (Review)

The software program you created using ioControl is called a *strategy*. In the previous lesson you saw that a typical ioControl strategy consists of the following:

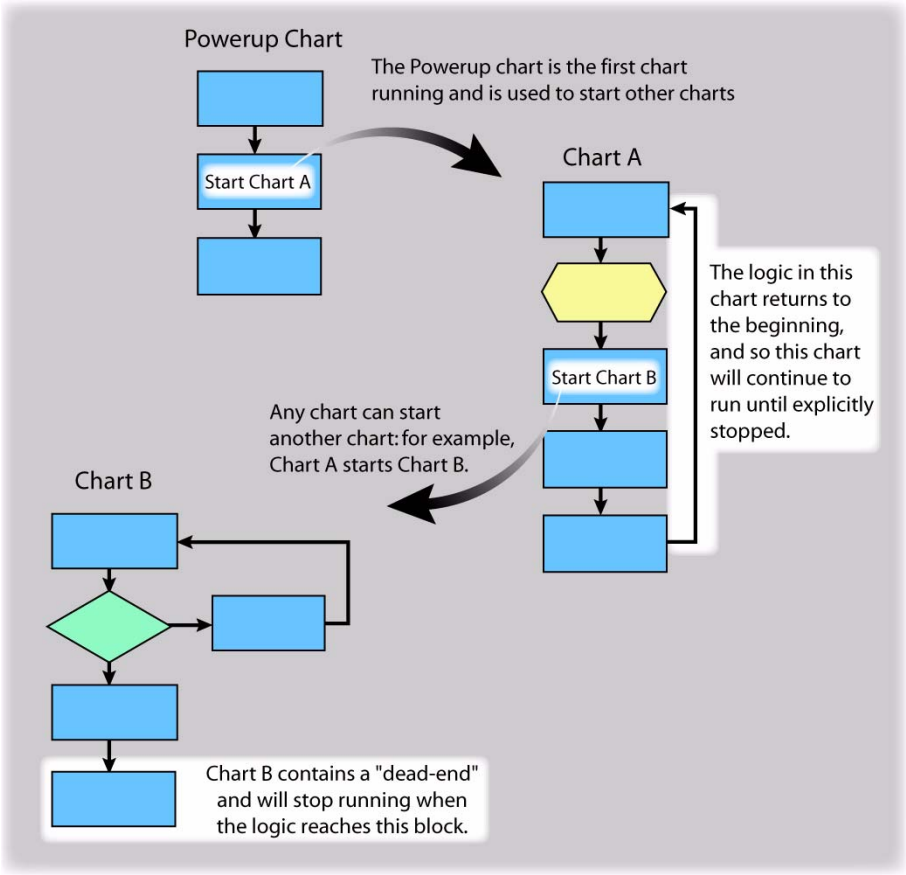
- Control Engine—This defines communication between the PC and the SNAP Ultimate I/O unit. It is required to download and run strategies in Debug mode. Your strategy contains the control engine you created in Lesson 2.
- Flowcharts—These contain the programming logic of your strategy. All strategies include a Powerup chart used to start the charts you create.
- Variables—These contain information as integers, floating points, strings, etc.
- I/O Configuration—This information defines the points that can be controlled. You recently assigned your I/O unit the loopback address of 127.0.0.1, ensuring that this strategy uses the local I/O unit on any SNAP Ultimate I/O unit it resides on.

Flowcharts

Since most control applications are complex, a strategy typically consists of several process flowcharts, or charts, that all work together. Each chart controls one aspect of the strategy—one piece of the automated process. Together, all the charts constitute the strategy. The total number of charts in a strategy is limited only by the amount of memory available in the control engine.

A chart can be running, suspended, or stopped. A running chart is actively performing its assigned task. A suspended chart is temporarily paused. A stopped chart is inactive. Every chart in an ioControl strategy can change the status of any other chart in the strategy, yet every chart is independent. Any combination of charts can be running simultaneously, up to the maximum of eight with SNAP Ultimate I/O (and 16 with the SNAP-LCE controller).

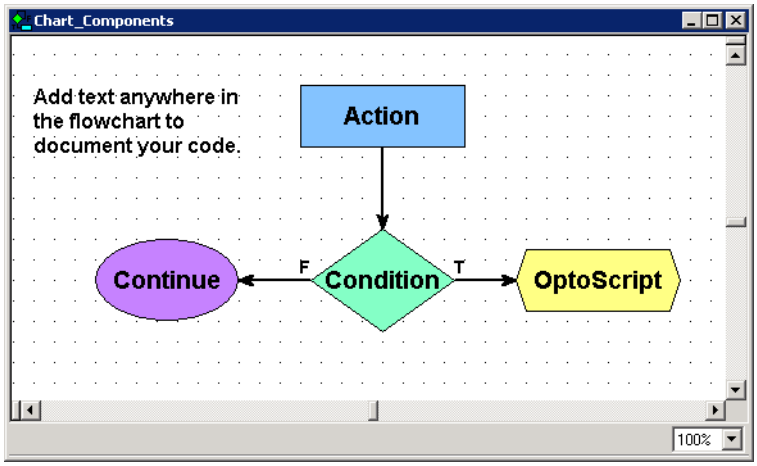
Every strategy contains a Powerup chart. The Powerup chart automatically starts when the strategy begins running, so it can be used to start other charts.

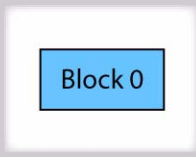


Blocks

Each chart is made up of blocks connected by arrows, which show how the process flows. Action blocks are rectangular in shape and indicate action within the process. Condition blocks are diamond-shaped and indicate a decision point. OptoScript blocks are hexagonal and contain OptoScript code, an optional method of programming. Continue blocks are oval and simply point to another block in the chart to continue the

process. Action, Condition, OptoScript, and Continue blocks are shown in the chart below:

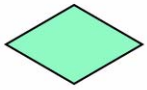




Block 0 is created for your when you create a new chart. A chart's programming logic always begins at Block 0, regardless of where it is within your chart. Block 0 can be renamed, but it can't be deleted.

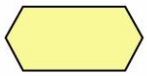
Action block

Condition block



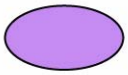
These blocks contain Action or Condition commands. When editing these blocks, you can select action commands or condition commands, depending on which type of block you are editing. If you insert multiple commands in an action block, the commands are executed in order. Multiple commands in condition blocks are evaluated collectively: for example, condition 1 **and** condition 2; condition 1 **or** condition 2.

OptoScript block



You can use the script-based equivalents of action and condition commands in script blocks, as well as a variety of logical structures.

Continue block

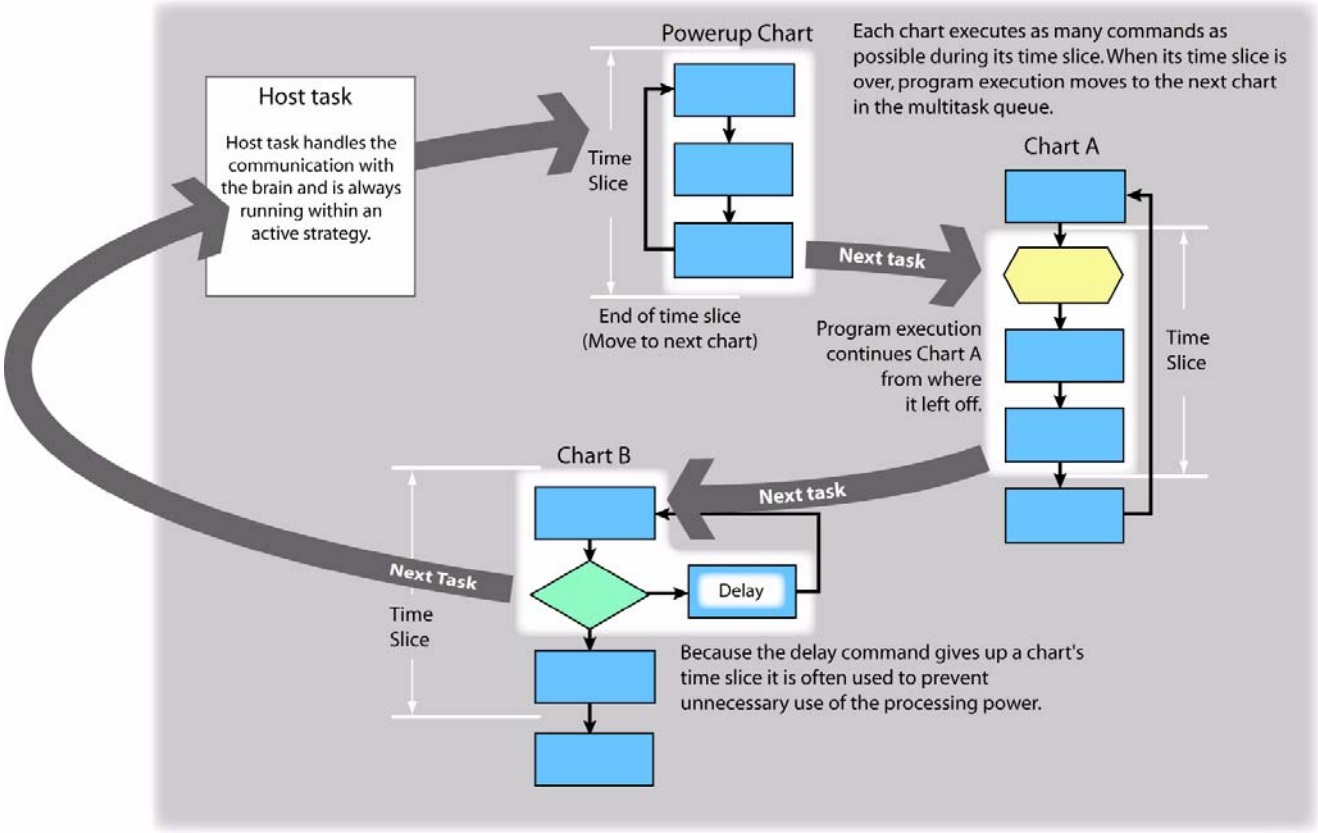


You can use Continue blocks as an alternative to drawing connection lines within a chart. As with any Go To command, use Continue blocks judiciously.

Multitasking

The control engine can run several charts seemingly at once, each performing a different task, through a technique called multitasking (also called multi-charting). The Opto 22 control engine contains multitasking firmware that allows it to run up to eight tasks (charts) simultaneously by assigning each task a time slice. (The SNAP-LCE control engine runs up to 16 charts simultaneously.)

The host task is an "invisible chart" used to communicate to a PC (which may be running ioControl in Debug mode or ioDisplay). Each chart in a running or suspended state is considered a task. Charts that are stopped are not counted in the eight simultaneous task limit.



Variables

A variable is a holding place that represents a piece of information in a strategy, such as the parameters for communication, temperature reported by a thermocouple, the name of a chart, or a group of words and numbers to be sent to a display. The information a variable represents is called the value of the variable. As a strategy runs, the variable's name remains the same, but its value may change. For example, the value of a variable named Fuel_Level may change several times while its strategy is running, but its name remains *Fuel_Level*.

A variable stores one of six types of data: floating point, integer, timer, string, pointer, or communication handle. When you create the variable, you designate the type of data it contains.

- A **Floating point** (or float) is a numeric value that contains a decimal point, such as 3.14159, 1.0, or 1234.2. A good example of a float variable is one that stores readings from an analog input, such as a thermocouple.
- An **integer** is a whole number with no fractional part. Examples of integer values are -1, 0, 1, 999, or -456. The state of a switch, for example, could be stored in an integer variable as 1 (on) or 0 (off).
- A **timer** stores elapsed time in units of seconds with resolution in milliseconds. Up timers count up from zero, and down timers start from a value you set and count

down to zero. For example, you could set a down timer to make sure a value is updated at precise intervals.

- A **string** stores text and any combination of ASCII characters, including control codes and extended characters. For instance, a string variable might be used to send information to a display for an operator to see. A string variable can contain numeric characters, but they no longer act as numbers. To use them in calculations, you must convert them into floating point or integer numbers. And a numeric value to be displayed on a screen must be converted into a string first.
- A **pointer** does not store the value of a variable; instead, it stores the memory address of a variable or some other ioControl item, such as a chart or an I/O point.
- **Communication handles** store parameters needed for communication with other devices.

You can use variables that are individual pieces of information, and you can also use table variables, which are groups of related information in the form of a table.

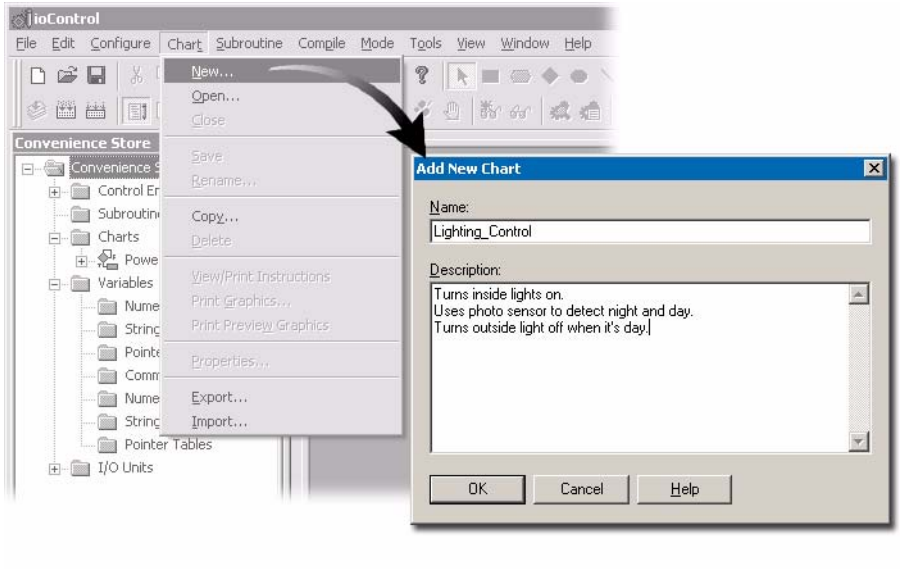
ACTIVITY

Creating a Flowchart

1. Create a new chart.

When you open a new ioControl strategy one chart already exists in the strategy: the Powerup chart. You create other charts as needed.

a. To create a new chart, select *New* from the *Chart* menu.



b. Type `Lighting Control` in the *Name* field. The space will automatically be replaced by an underscore.
c. Press the *Tab* key on the keyboard to advance to the next field.
d. Type an explanation of the chart's function in the *Description* field.
e. Click *OK*.

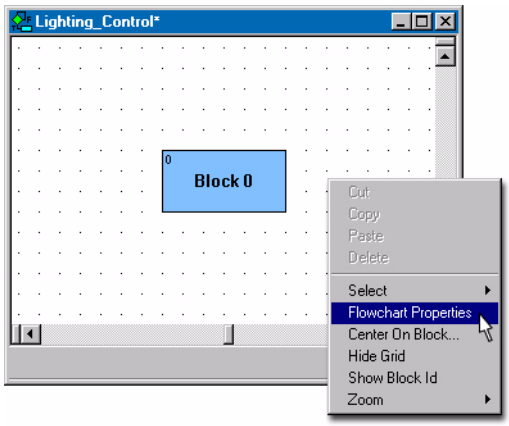
An action block called *Block 0* is the only block in the new chart. Block 0 is always the first block executed in any chart. Block 0 can be renamed, but it cannot be deleted.

Choosing Flowchart Colors and Text

1. Change flowchart properties.

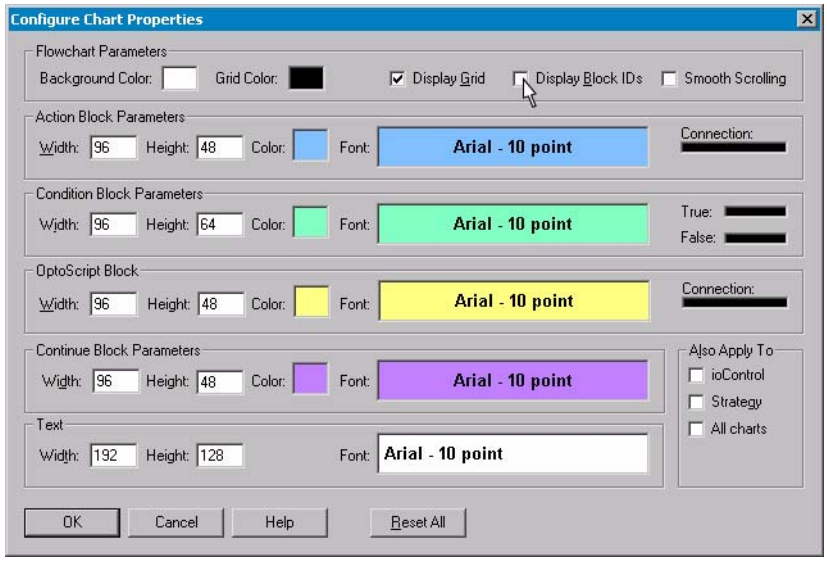
Many features of charts are available in context-sensitive menus.

a. Place the cursor in the *Lighting_Control* window.
b. Click the right mouse button.
c. Select *Flowchart Properties* from the shortcut menu.

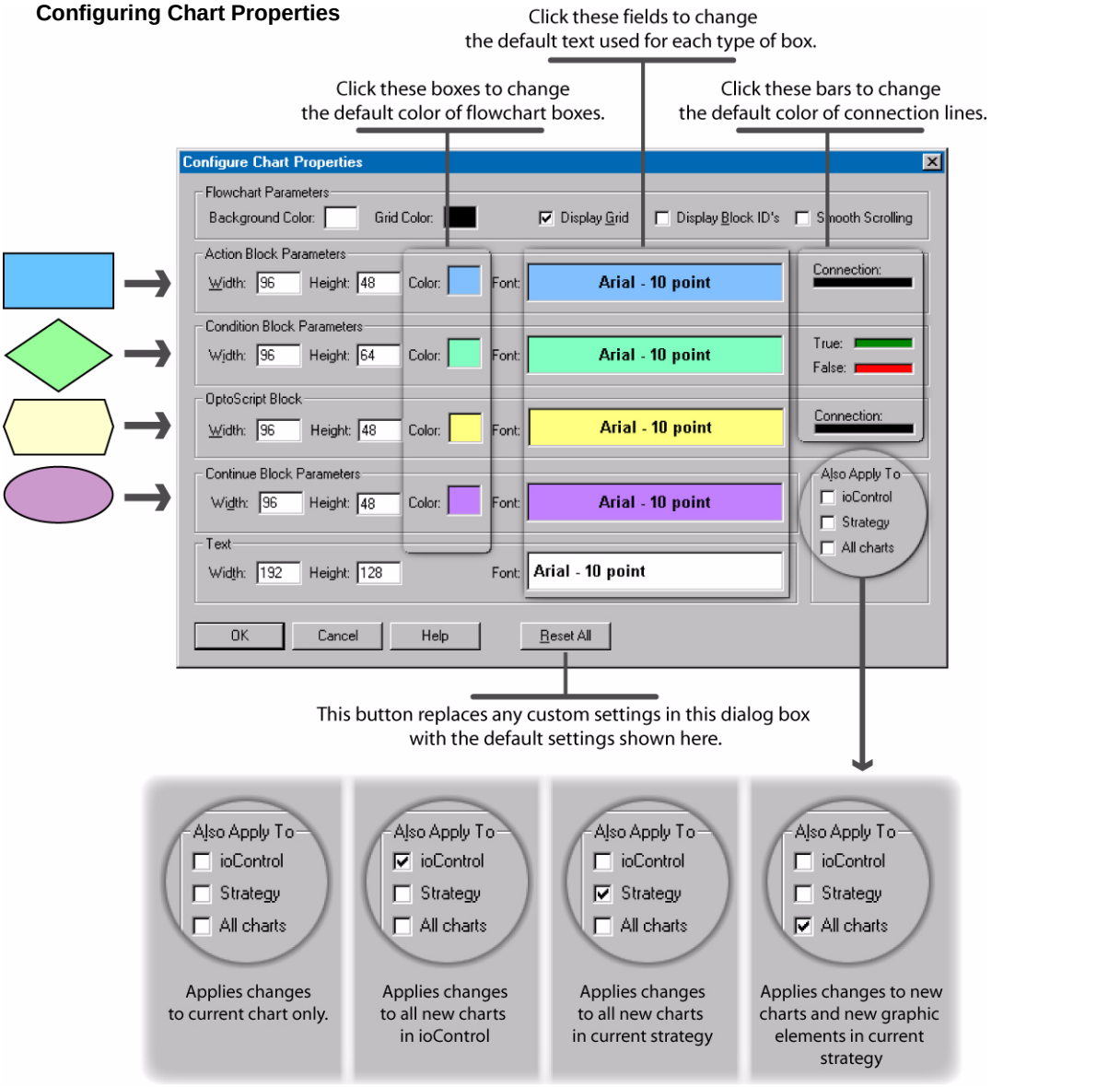


2. Change the default color of Action Blocks.

a. Click the *Display Block IDs* box to uncheck this box and turn off the ID number display.

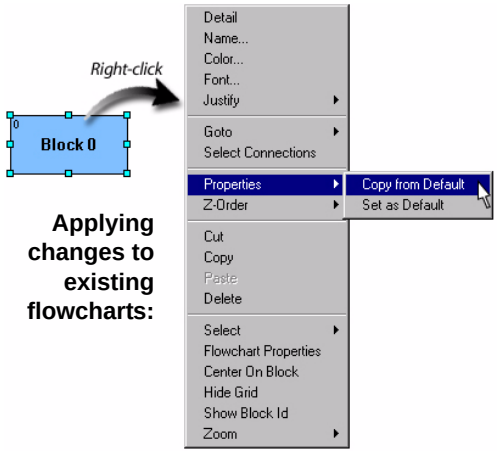


Configuring Chart Properties

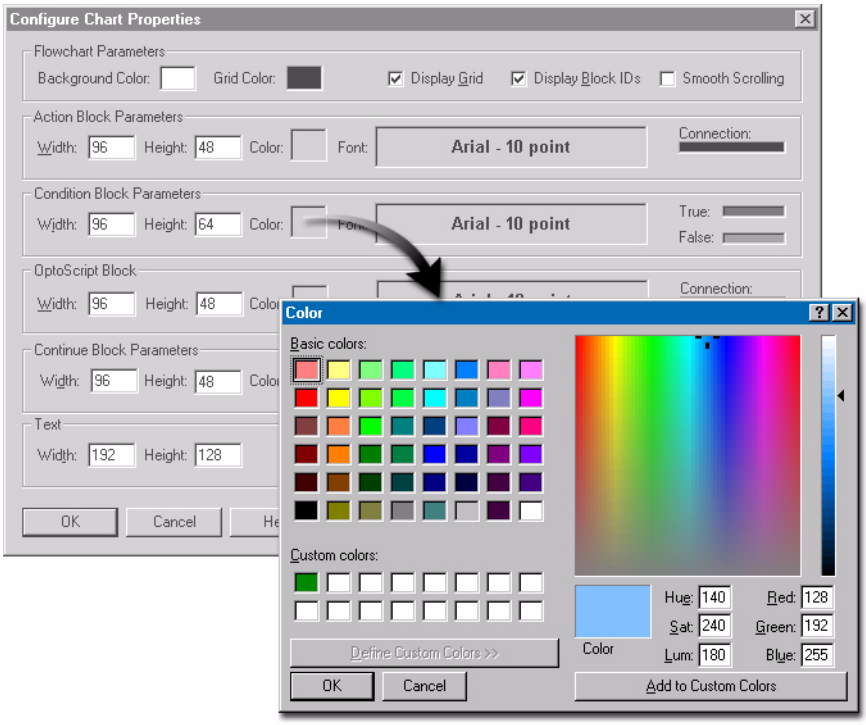


From the *Configure Chart Properties* window, you define the default size and color of any flowchart object or text. You can also define the flowchart background and grid color. The block ID numbers and grid can be turned on or off from this window. Block ID numbers are displayed in the upper-left corner of every action, condition, and continue block. Every block created in an ioControl chart is given a unique ID number for reference. Once a number is used, it is not reissued even if the block is deleted. Block IDs are useful for differentiating between blocks which may have the same name (since block names don't have to be unique in ioControl). This could come in handy if you wanted to choose a block as the destination of a continue block, but have several blocks with the same name.

Applying changes to existing flowcharts:



b. Click the *Color* block in the *Condition Block Parameters* section.

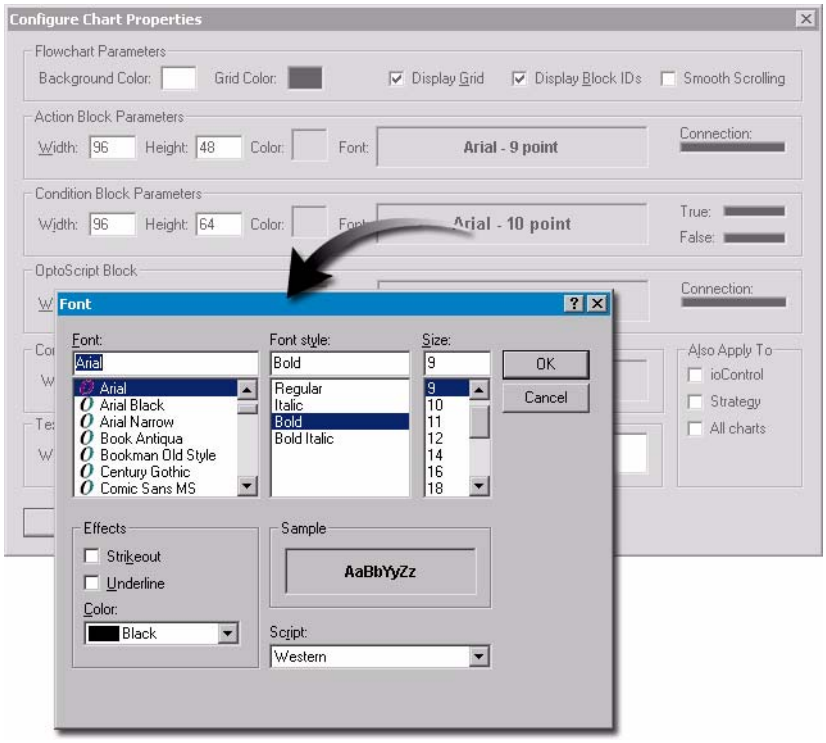


c. Click one of the color blocks displayed under *Basic Colors* to select a new default color of a condition block. (You can keep the existing color, if your prefer.)

d. Click *OK* to close the Color dialog box.

3. Change the default text of Condition blocks.

a. Click the *Font* box in the *Condition Block Parameters* section.



b. Enter *9* in the *Size* field to change the text font for condition blocks. Feel free to change the font or color as well.

c. Click *OK* to close the Font dialog box.

Experiment and change the colors and fonts of the *Action* and *Continue Blocks*.

You can also change the color of the connection lines by clicking on the narrow rectangular block under *Connection*, *True Connection* and *False Connection*. Some ioControl users like to make the true connection line green, and the false connection line red.

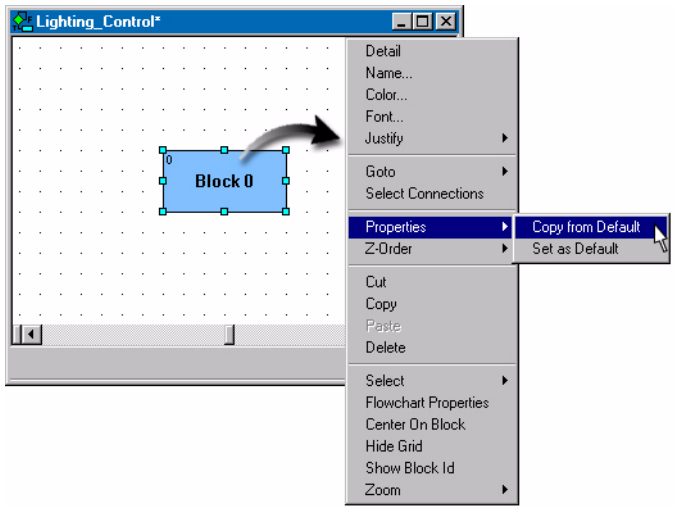
4. Apply Your Changes.

The *Also Apply To* options expand the scope of the chart properties. Selecting *ioControl* applies the changes throughout all new strategies. Selecting *Strategy* applies the changes throughout all new charts in the current strategy. The *All Charts* option applies the changes to all new charts and new graphic elements in the current strategy.

a. Click the *ioControl, Strategy,* and *All Charts* options in the *Also Apply To* field to apply these defaults to all new charts in this strategy and future strategies. (Changes take effect as new charts or new chart objects are created.)

b. Click *OK*.

Because it was created before you changed the default properties, Block 0 still has the old properties. To give Block 0 the new properties, select the block, right-click, and select *Properties* ⟶ *Copy from Default*.



Object properties can be changed individually to easily customize the look of your flowchart. The properties for individual objects can be changed by selecting the object, right-clicking, and choosing the property to be changed from the pop-up menu.

5. Reset the Changes

If you do not wish to use the new settings, do the following:

a. Open the Configure Chart Properties dialog box.
b. Choose *Reset All*.
c. Click *Apply*.

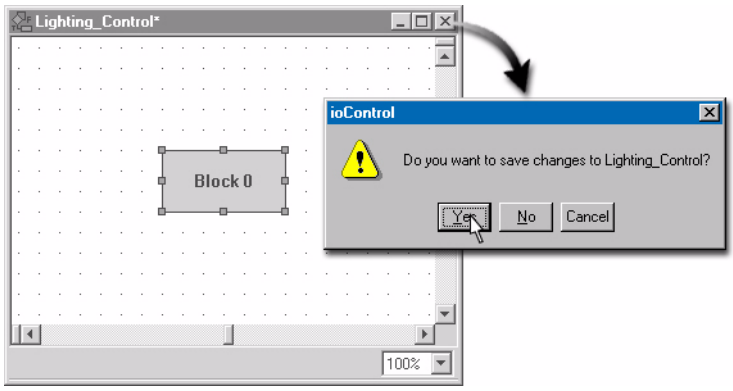
Closing Your Chart

You will close your Lighting_Control chart for now.

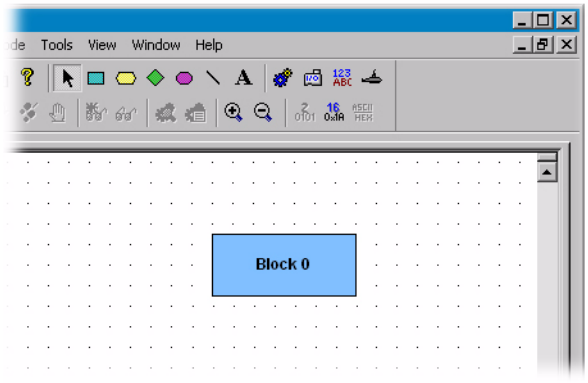
a. Close the Lighting_Control chart by clicking the close box (×) in the upper-right corner of this window.

   When you close a chart window where you have made changes, you are prompted to save the chart.

b. Click *Yes* to save changes to your chart.



Close Box on Maximized Charts



If you have maximized your chart, you will find the close box for the chart in the top right corner of the ioControl application window.

Adding a Command to the Powerup Chart

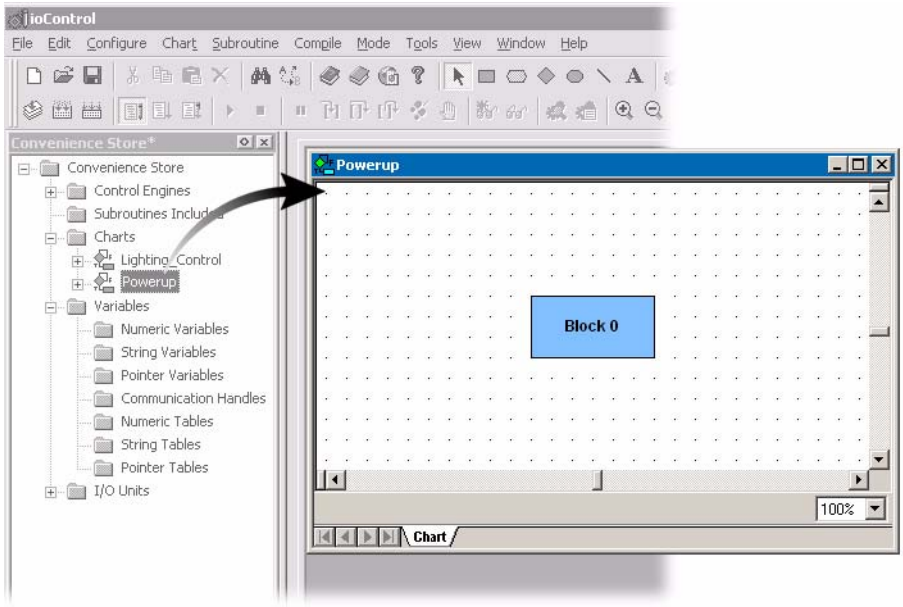
You've created the chart Lighting_Control. For this chart to run, it must be started by a command in another flowchart. The Powerup chart is started automatically, so this is the best location for commands that start all charts you want running when the strategy starts.

To create this capability you will do the following:

- Add an Action block to the Powerup chart.
- Add a Start Chart command to the action block.
- Create a variable to contain the command to start the Lighting_Control chart.

1. Open the Powerup chart.

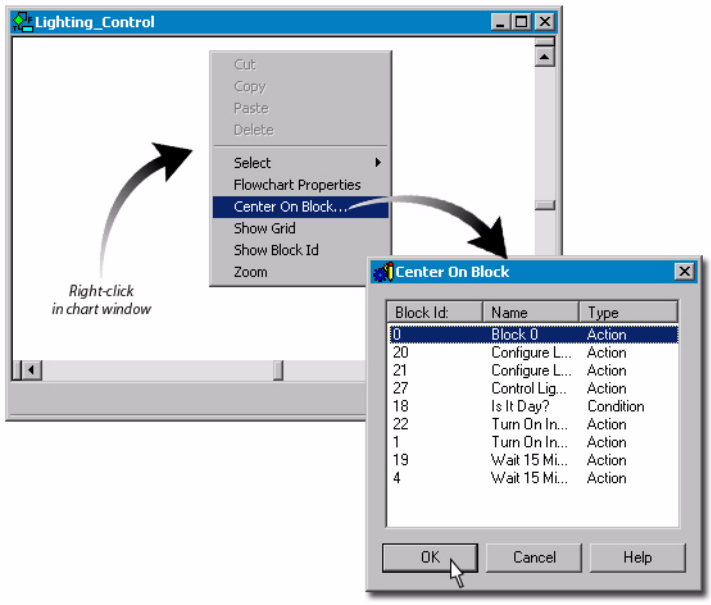
a. Double-click *Powerup* in the Strategy Tree.



b. Resize the chart if desired. You can maximize the window by clicking the maximize button ( )
c. Use the scroll bars at the bottom and left of the window to center Block 0 in the Powerup window.

If you can't see your chart objects:

If you scroll too far, you can lose sight of your flowchart objects. Use Center on Block to locate any chart block:



1. Maximize your chart.

2. Right-click in a chart window.

3. Select *Center on Block*.

4. In the Center on Block dialog box, select a block, for example, Block 0.

5. Click *OK*.

(The example shown here is based on a completed flowchart you create in this lesson.)

2. Assign new properties to the Powerup chart

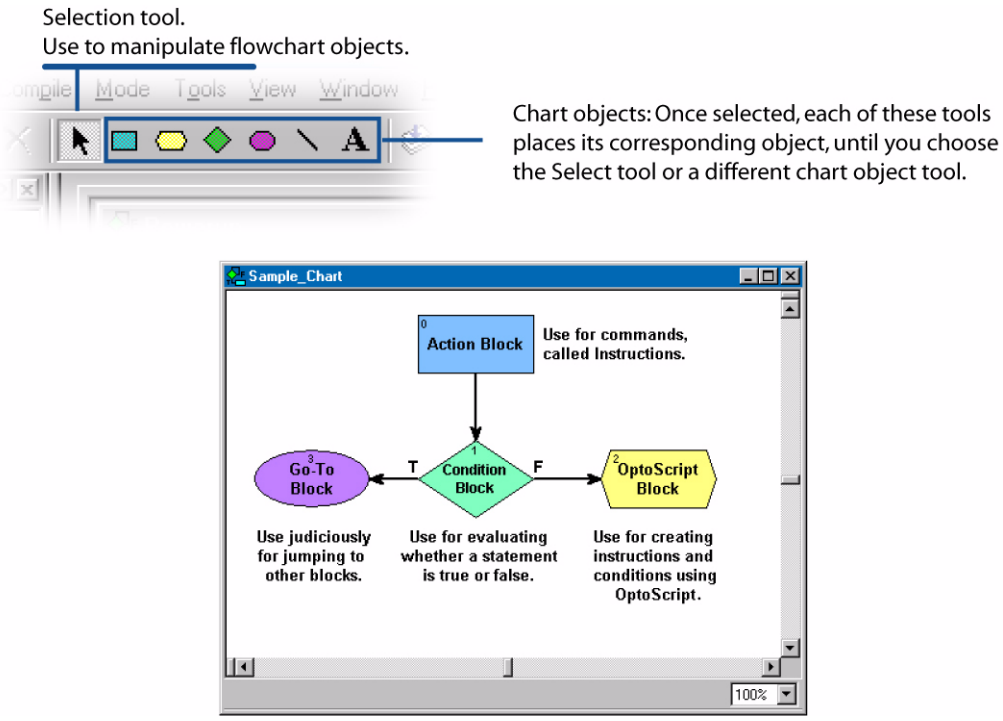
Because the Powerup chart was created before you changed the default chart properties, Block 0 still has the old properties.

To give Block 0 the new properties, select the block, right-click, and select *Properties* ➞ *Copy from Default*.

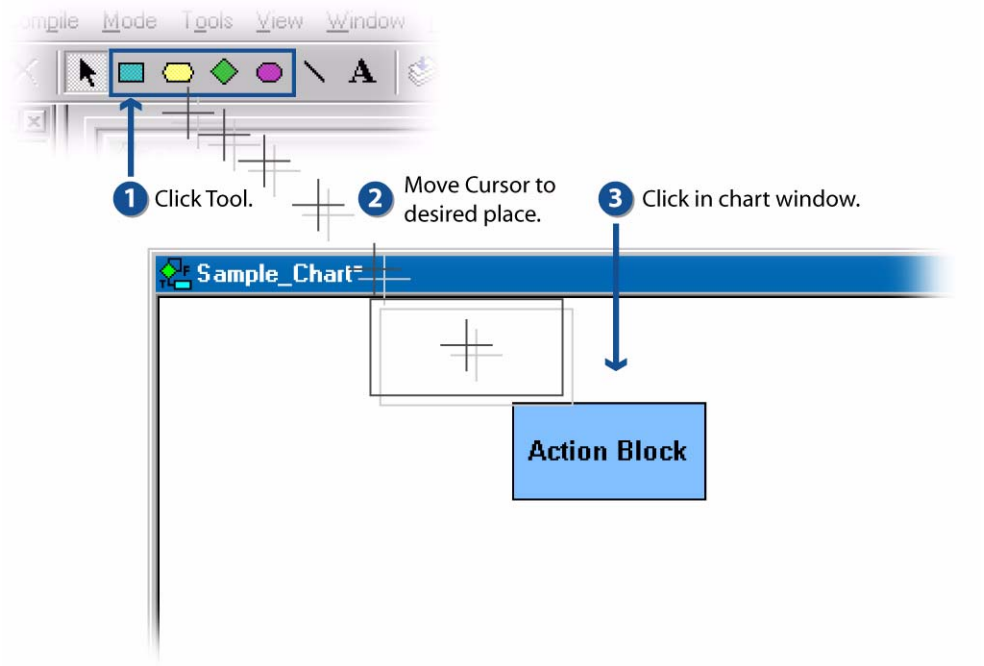
Using the Flowchart Drawing Tools

There are six drawing tools in ioControl. The drawing tools are used to create flowcharts and select flowchart objects.

Chart Tools:

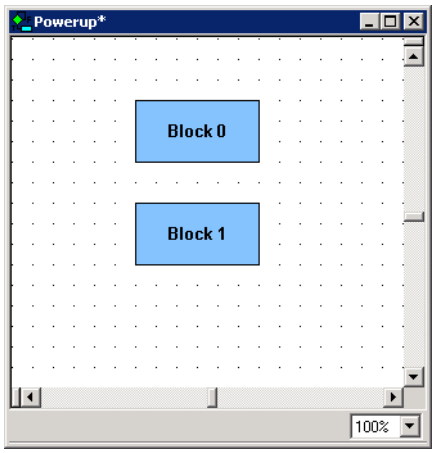


Drawing New Blocks:



1. Draw an Action block.

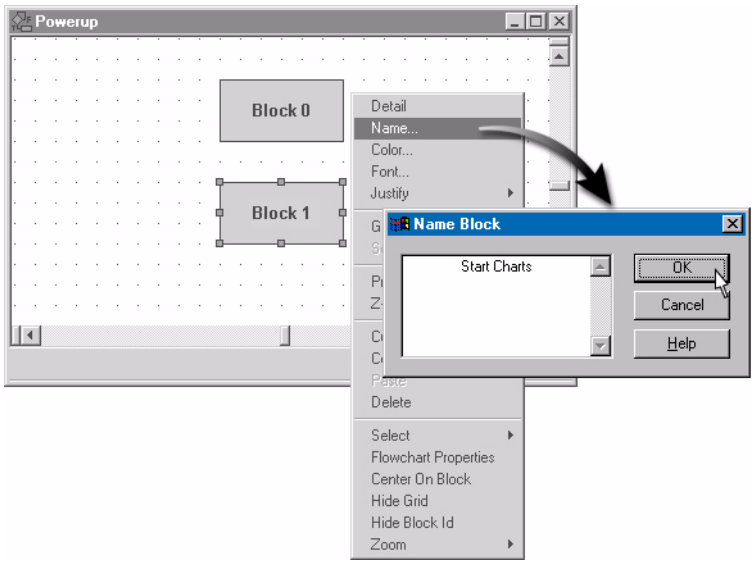
a. Select the *Action Block Tool* from the tool bar.
b. Position the cursor below Block 0 in the Powerup chart.
c. Click the left mouse button.



Block 1 has been created in your flowchart. *Block 1* is the default name of this action block. Every block created in ioControl is given a default name. Once a name has been issued, it is not reissued. If you delete Block 1 and draw a new block, the name of the new block will be *Block 2*. The default names of the blocks are temporary. You should always change the name of a block to describe the function of the block in the flowchart.

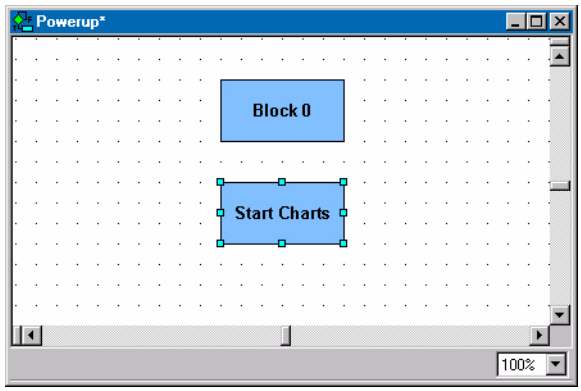
2. Rename an Action block.

a. Click the Select tool ( ).
b. Click Block 1 to select it.
c. While Block 1 is selected, click the right mouse button. A pop-up menu of available options for the selected object opens.
d. Select *Name* from the list.



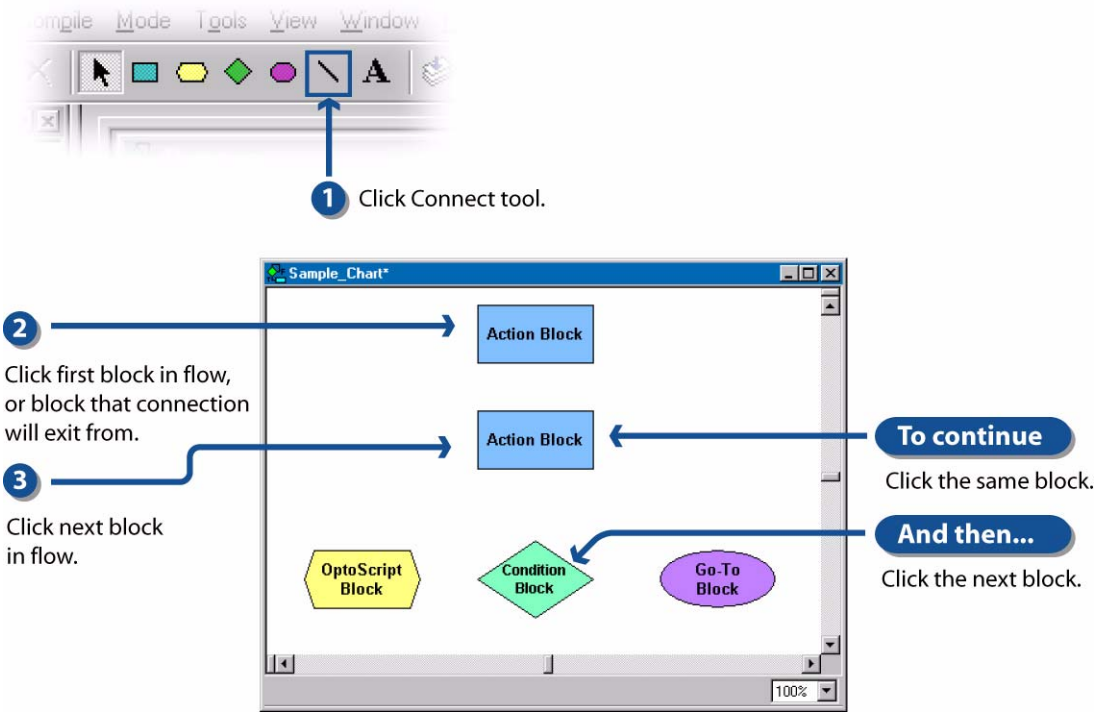
e. Type `Start Charts` as the name of the block.

f. Click *OK*.



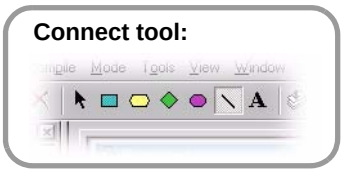
Now, you will draw a line defining the flow of logic in the chart.

Connecting Blocks:

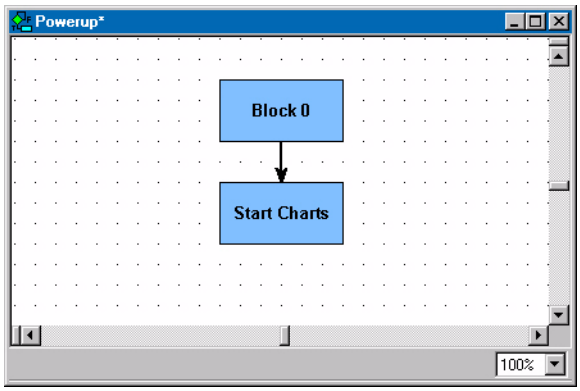


3. Connect Start Charts block to flow.

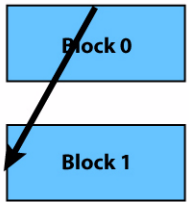
a. Select the Connect tool.
b. Click *Block 0*.
c. Click *Start Charts*.



d. Click the Select tool to release the Connect tool. (You can also press the Escape key, or right-click on a blank area in the flowchart window to release the Connect tool.)



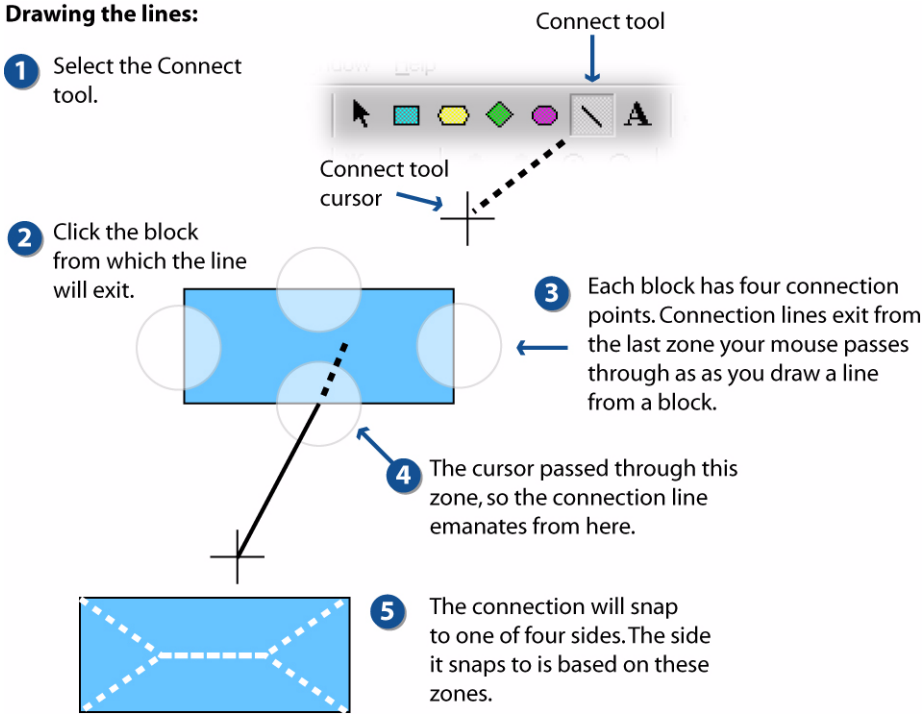
Controlling Your Connection Lines:



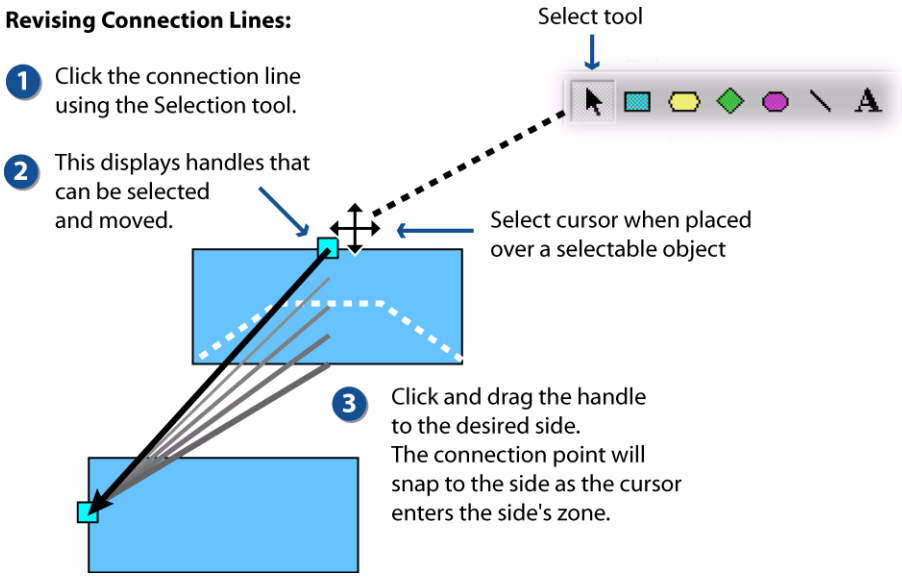
Do your connection lines look like this?

If you are having difficulty using the connection tool, the diagrams below provide some advice on how to draw and revise connection lines.

Drawing the lines:



Revising Connection Lines:

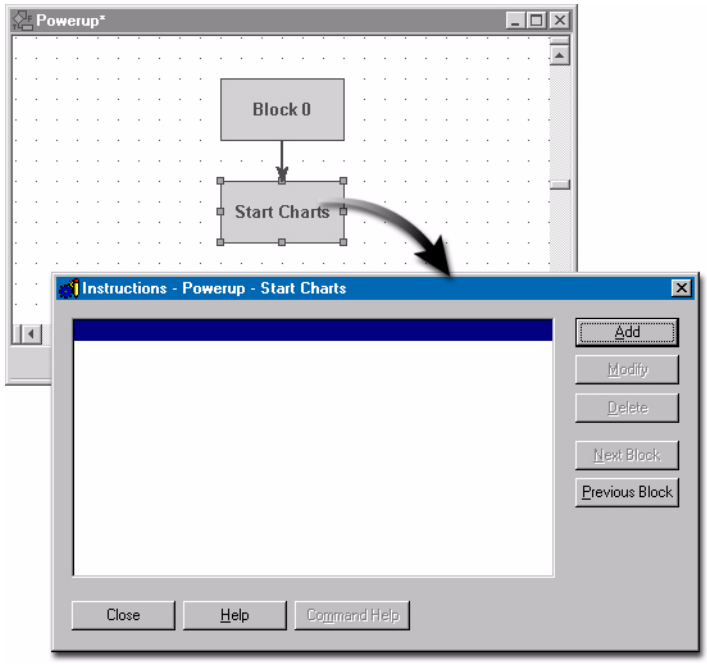


Adding an Instruction

The blocks in charts are just shells used to contain instructions and scripts. You've just placed a new action block and connected to the flowline, but this block must include some instructions.

1. Open the Instruction dialog box.

Double-click the *Start Charts* block to open the instruction window.

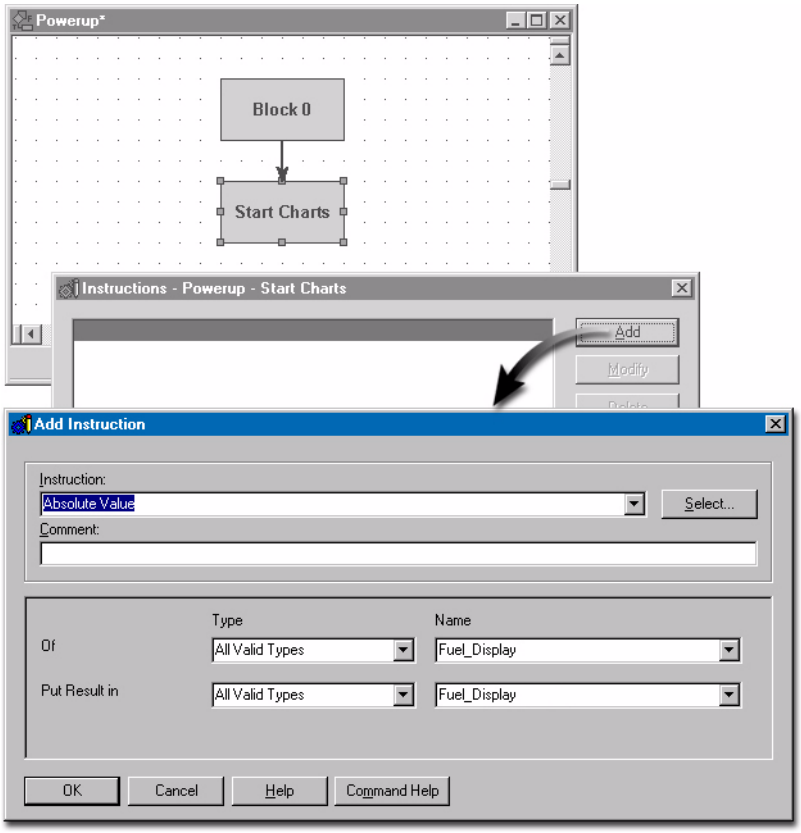


You can also open the instruction window by selecting the block, clicking the right mouse button, and selecting *Detail* from the pop-up menu

NOTE: Double-clicking a block opens the Instructions window. Other block properties, such as name and color, are changed by right-clicking the block and selecting from the context-sensitive menu.

2. Open the Add Instruction dialog box.

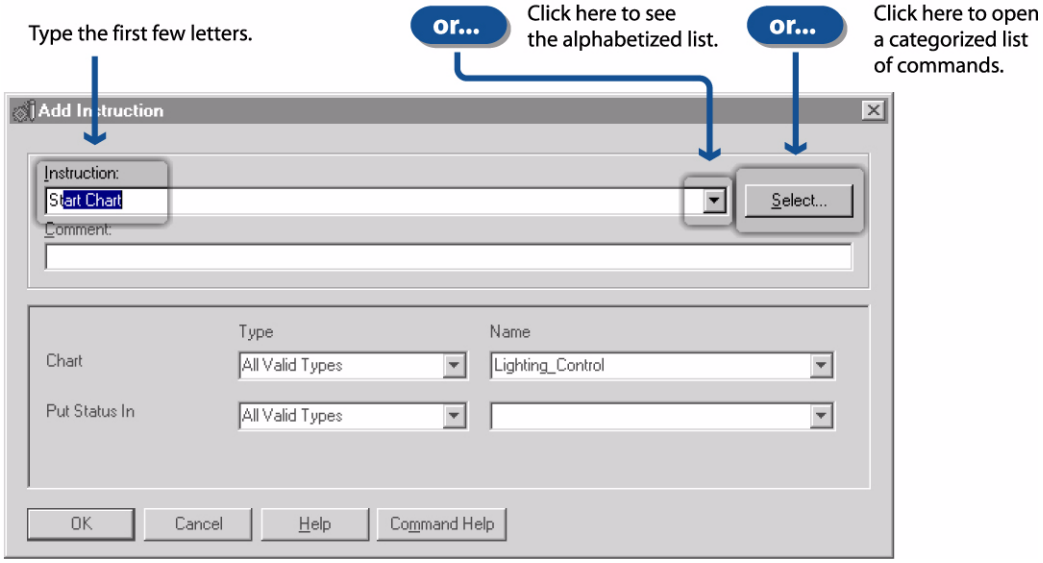
Click *Add* to enter a new instruction.



3. Locate the Start Chart instruction.

There are several ways you can select an instruction. Three methods are listed below. Select one of the methods to enter the Start Chart instruction into the Start Charts block.

Three ways to locate an Instruction:



One: Type the beginning letters of the instruction.

- If you know the instruction that you want, type the first few letters of the instruction until the instruction appears in the field. For the instruction *Start Chart*, you can type `St` and the *Start Chart* instruction is selected in the field.

Two: Use the drop-down button.

- You can click the drop-down button at the end of the *Instruction* field and scroll through the list of all available instructions. To quickly jump to different areas of the list, type a few letters in the *Instruction* field.
- Once you are in the drop-down field, you can use the up and down arrow keys on your keyboard to move through the list until you locate the Start Chart command.

Three: Use the Select button.

- If you do not know the name of the instruction you need, click the *Select* button.
- From the *Select Instruction* window, select the group that is related to the type of instruction desired. To find the Start Chart instruction, highlight *Chart* in the *Groups* list. *Start Chart* can now be selected from the list of instructions.

Completing an Instruction

Each instruction requires additional information to be complete. When you choose the Start Chart instruction, you must also provide the following:

- Name of chart to start. Lighting_Control already appears because it is first in an alphabetical list of available charts (the only other chart is Powerup).
- Variable to contain status of loading the chart.

When a Start Chart instruction is executed, ioControl returns a status of 0 if the requested chart was successfully started. If the chart was already running or unable to start because eight tasks were already running, ioControl returns a -5 status. The status can be used to verify that a chart was successfully started before the flowchart strategy continues. To store this information, you'll need to create a variable.

NOTE: Status and error codes are described for each command in the command help, available from the *Command Help* button.

Defining variables:

When a feature such as an instruction requires a variable, you can select an existing variable or define a new one. If you define a new variable, the Add Instruction dialog box links to Variable Definition dialog boxes. This is called creating a variable on-the-fly.

As an alternative, you can always open the Variable Definition features from the Variables folders in the Strategy Tree.

It is always a good idea to select the type of variable that is appropriate for the type of data that will be stored. ioControl can store an integer in a floating point variable and let you use it as an integer. However, this requires an unnecessary data conversion. The unnecessary data conversion is automatically performed, but it does require processor time.

1. Provide a comment.

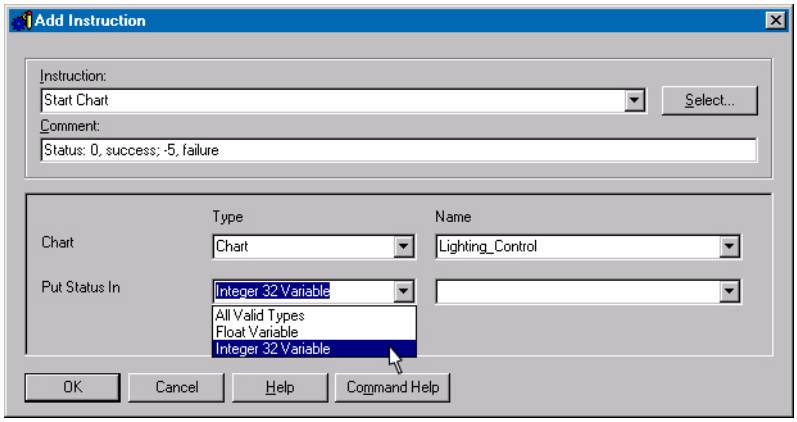
The Start Chart command name is self-describing, so in the comment field provide the following description (these are the status codes returned for this command):

```
Status: 0, success; -5, failure
```

2. Choose variable type.

The chart that you want to start, Lighting_Control, is already selected because it happens to be the first entry in the alphabetical list of charts in your strategy. If, for example, you had a chart named Analog_IO_Data, this chart would appear by default, and you would have to select the Lighting_Control chart instead. When the chart is started, a status code of 0 or -5 is put in the variable you determine. Since 0 and -5 are integers, use an integer 32 variable.

a. Select *Integer 32 Variable*.



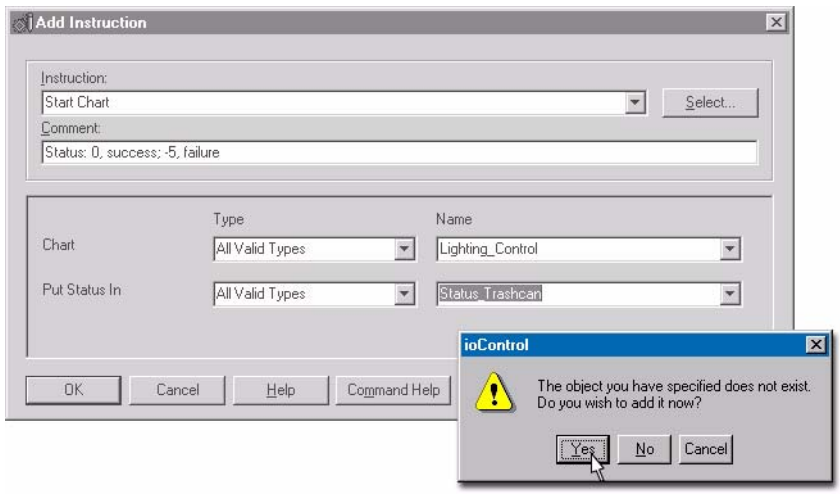
3. Define a variable.

a. Type `Status_Trashcan` in the *Put Status in–Name* field to create this variable using the on-the-fly method of variable configuration.

The spaces between words you type are automatically replaced by underscores. This new variable will store the value of the returned status.

b. Click the *Tab* key on the keyboard.

Since Status_Trashcan is an undefined variable, you are prompted to add the variable to your strategy.

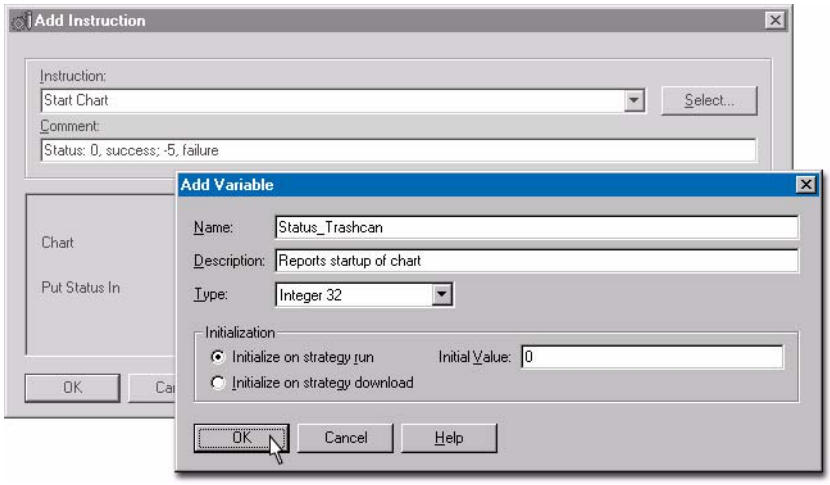


c. Click *Yes* to add the new variable.

The Add Variable dialog box is displayed.

d. Type a description of the variable's function:

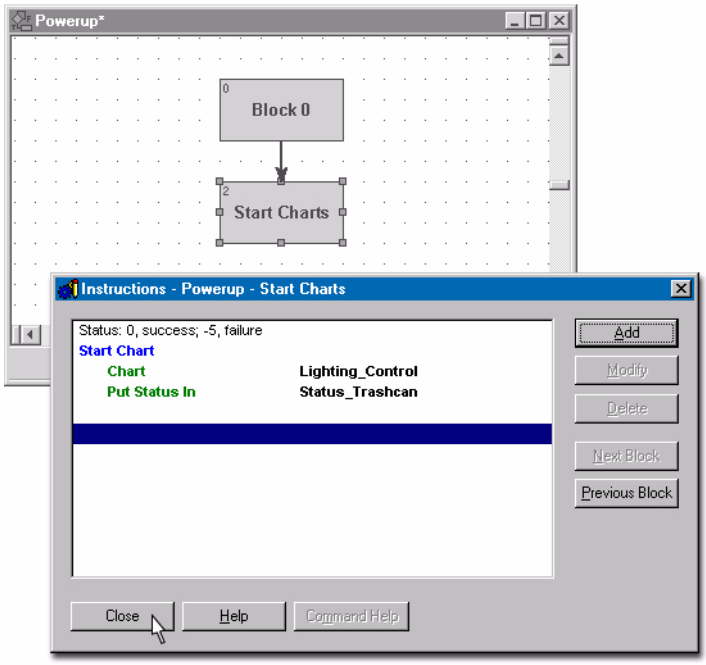
`Reports startup of chart`



e. Click *OK* to close the Add Variable dialog box.

f. Click *OK* to close the Add Instruction dialog box.

The completed instruction can be viewed in the Instructions window.



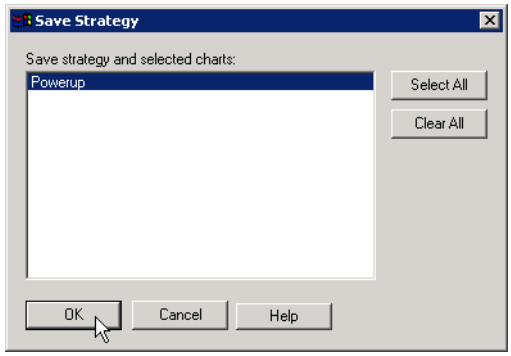
g. Click *Close* to complete the entry of this instruction.

Additional instructions can be created, which you will do later when you start another chart. When you have multiple instructions within a block, instructions are executed in listed order. The order can be changed by using Copy and Paste to move the commands.

4. Save the strategy.

a. Select *Save Strategy* from the *File* menu.

Modified charts that have not been saved are listed in the Save Strategy window.



b. Click *OK* to save the highlighted charts.

In general, it is a good idea to save your strategy every 10 minutes. When a strategy is downloaded to a control engine, it is automatically saved. Therefore, if you are frequently downloading and testing your strategy, it is not necessary to continuously save your strategy.

Archiving Your Strategy

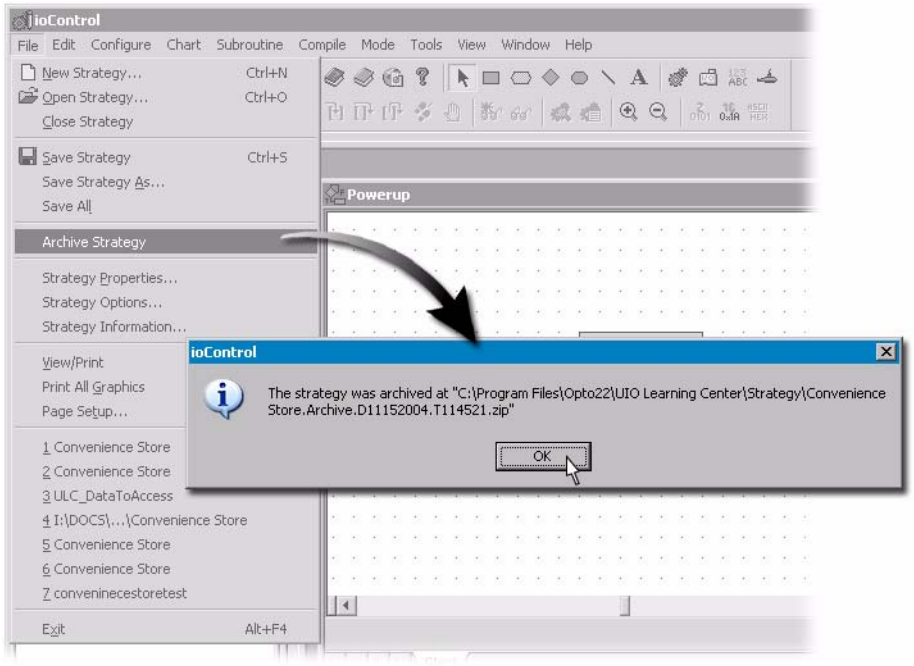
It is a good idea to save several copies of your strategy as it is developed. This practice makes it easy to revert to previous versions of your strategy.

Archive Strategy creates a point-in-time snapshot of your strategy within a ZIP file. Archiving your strategy has three main advantages:

- You can store multiple archives in the same directory.
- Archiving stores all related files as one ZIP file.
- The archive file has a time and date stamp in the file name.

1. Create an archive.

a. Select *File* ⟶ *Archive Strategy*.

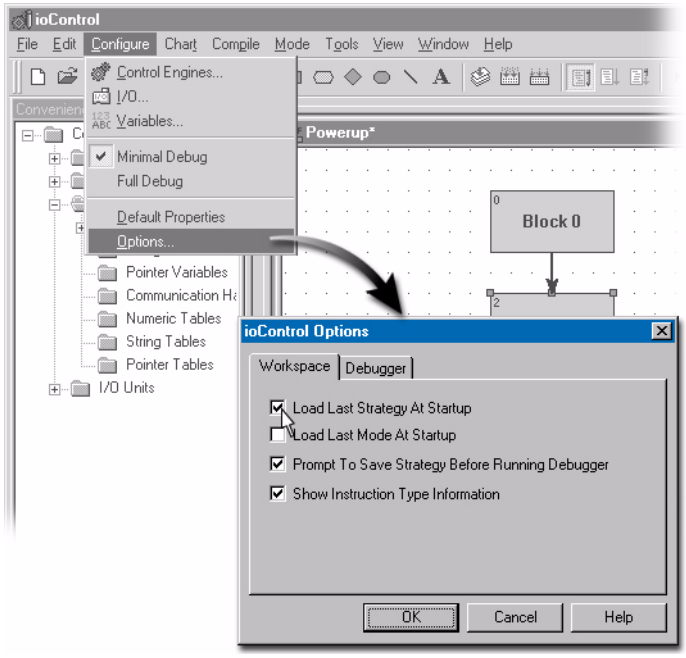


b. Click *OK*.

Setting Workspace Options

You can configure ioControl to automatically open the last strategy that was opened.

1. **Select** *Configure* ⟶ *Options*.



2. **Check the box** *Load Last Strategy At Startup*.

3. **Click** *OK*.

Leave your strategy and ioControl open if you plan to continue to Lesson 4; otherwise, save and close your strategy and close ioControl.

DIGITAL CONTROL

SKILLS

Configuring a Strategy

- Building the Chart
- Adding Control for the Freezer_Door and the Emergency Switch
- Modifying the Lighting_Control Chart

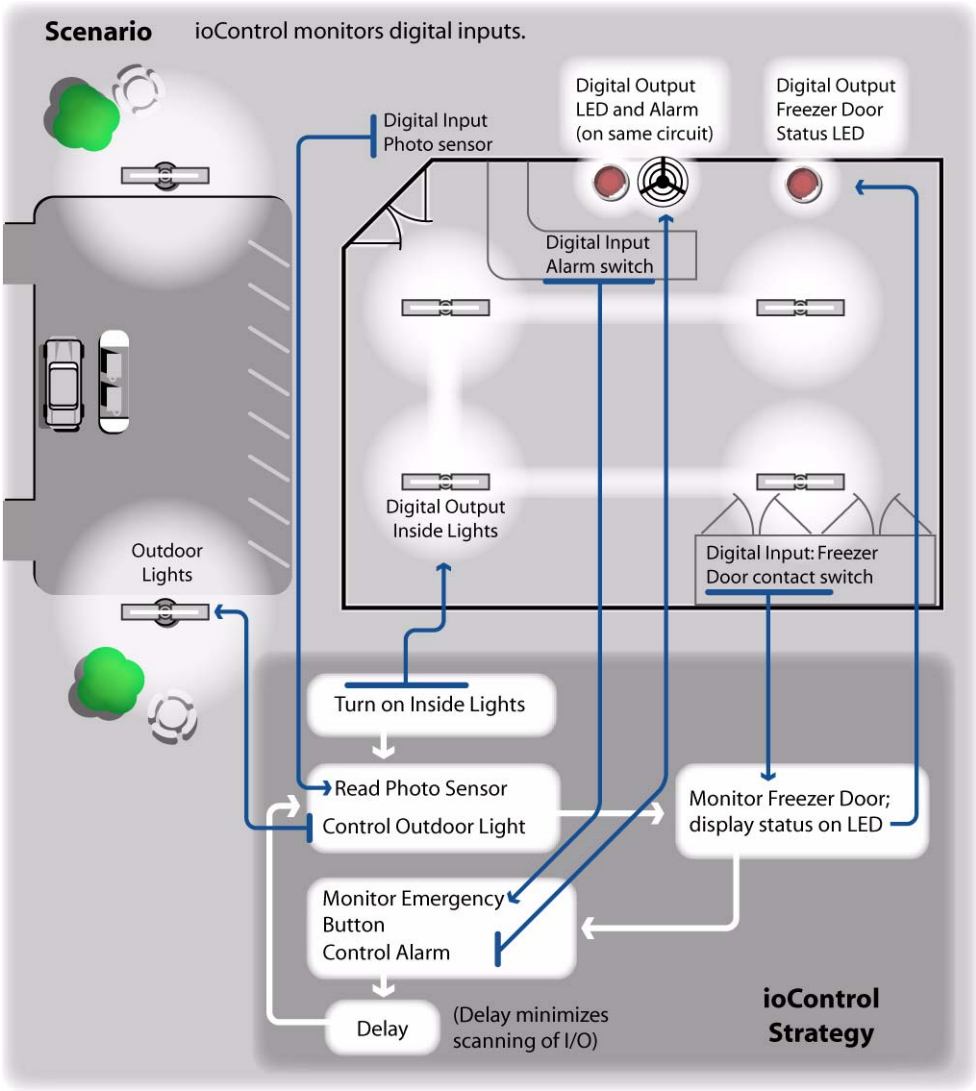
Debugging

- Configuring Debug Mode
- Downloading and Running the Strategy
- Using Auto Stepping
- Using Step Buttons
- Debugging the Changes
- Forcing Outputs and Inputs in Debug Mode

SCENARIO

In the previous lesson, you started building your programming logic by creating a chart to control the store's lighting and started this chart by adding commands to ioControl's built-in Powerup chart. Though your Lighting_Control chart runs, it lacks logic to control anything. In this lesson you will add logic that:

- Turns on the inside light.
- Reads the photo sensor and uses its state to control the outside light.
- Reports the state of the Freezer door through the Freezer Door Status light.
- Reports the state of the Emergency switch through the Alarm light.



ACTIVITY

Building the Chart

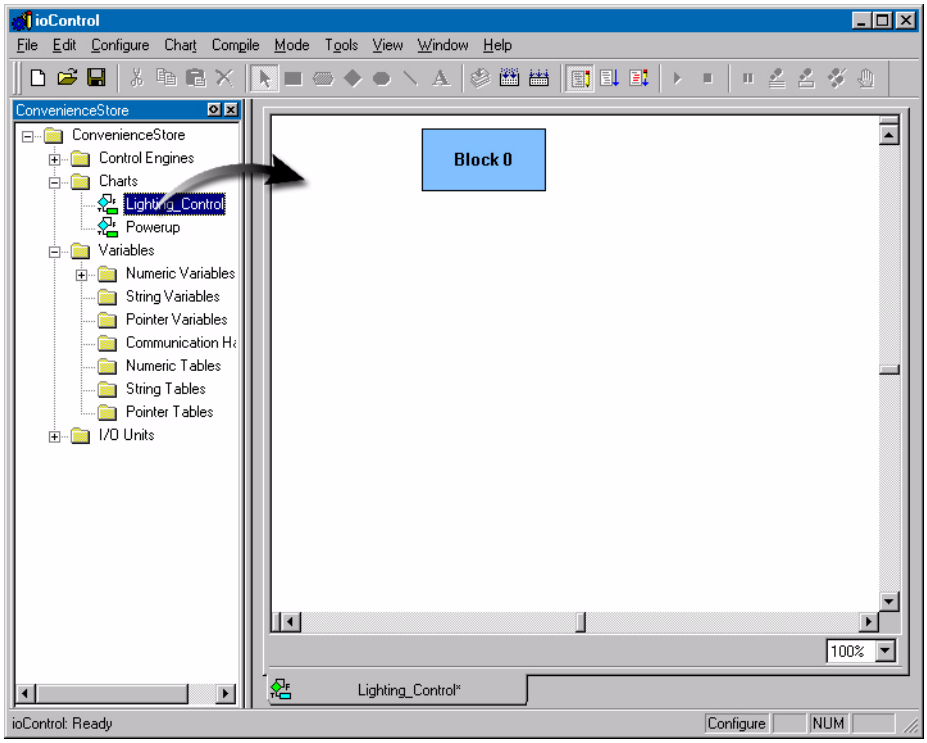
1. Open your strategy.

Open **ConvenienceStore.idb** in ioControl, if it is not already open.

2. Open your Lighting_Control flowchart.

a. Under the Strategy Tree, expand the Charts folder.

b. Double-click *Lighting_Control*.



c. Adjust the window if necessary.
d. Change any preferences, if desired.

Notice that in the example above and used throughout, the following has been done:

- The Strategy Tree has been docked (dock the Strategy Tree by clicking the dock button ( ).
- The grid and block numbers have been turned off using the Configure Flowchart Properties dialog box.
- The flowchart window has been maximized.

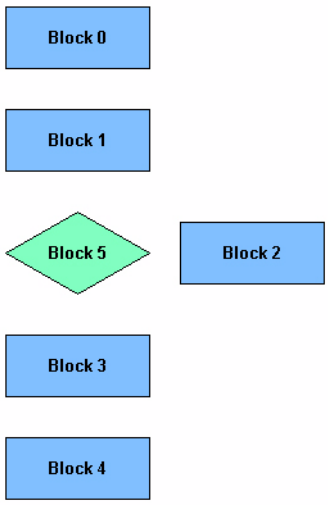
Review: Configuring Flowchart Properties

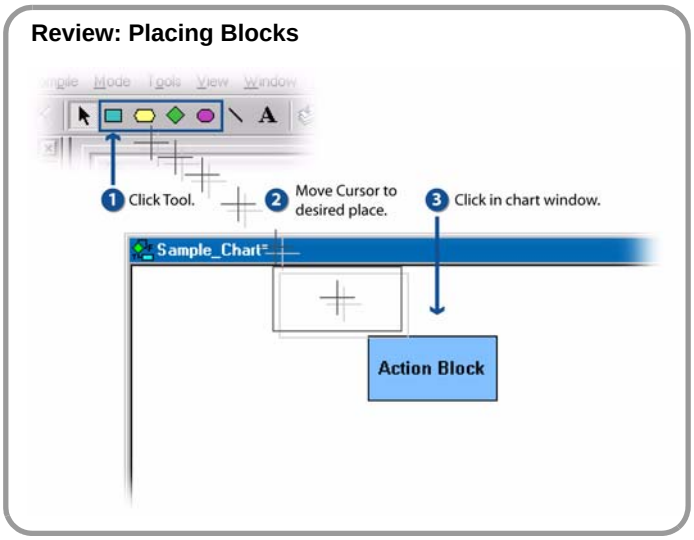
1. Right-click the flowchart window.

2. Click Flowchart Properties.

3. Add Action and Condition blocks.

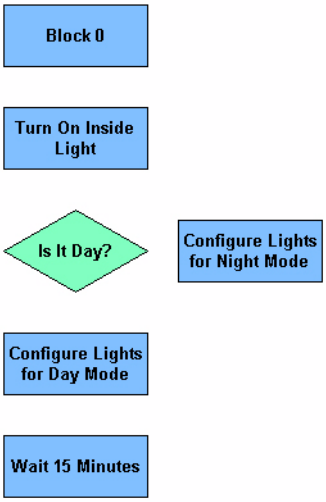
Add Action and Condition blocks so that your flowchart appears as shown below:

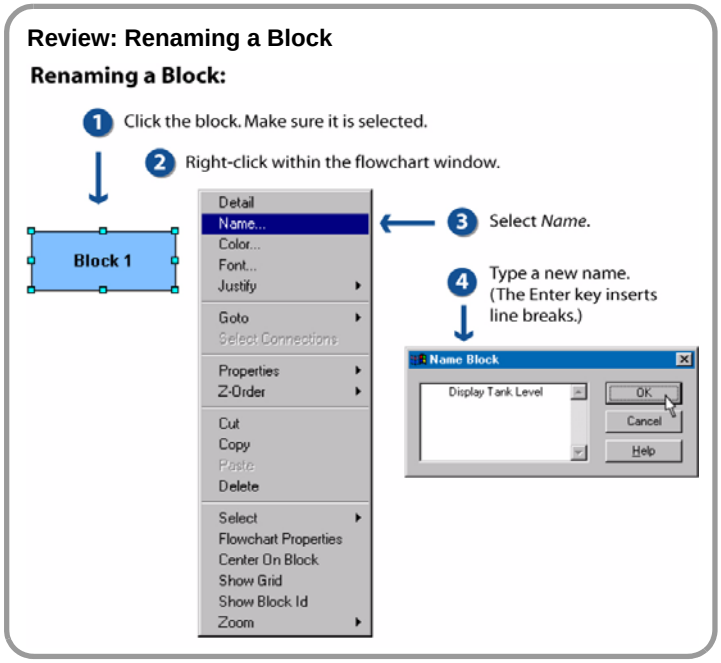




4. Rename blocks.

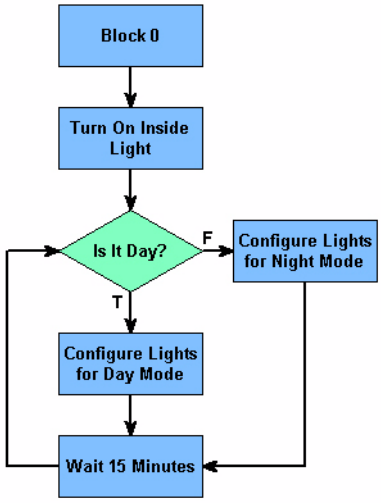
Rename each block as shown below:





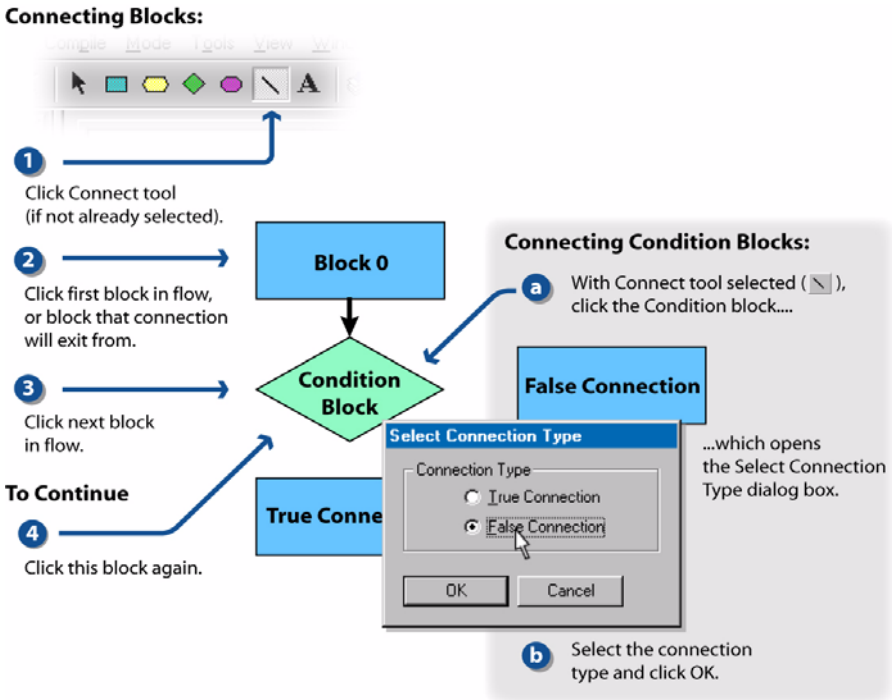
5. Connect the blocks.

Connect the blocks, as shown below:



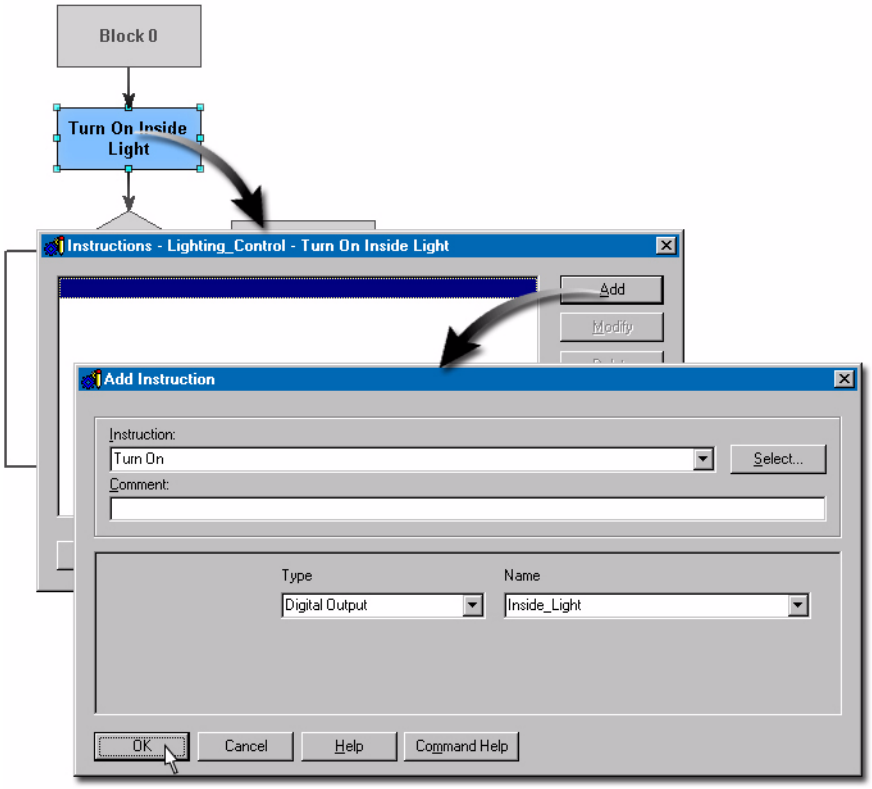
TIP: You can create corners with the Connect Tool. While drawing a connection line, any click on the chart window (not on a chart object) will place a corner. Each time you click the mouse, ioControl will anchor the connection line. If you have not completed the connection line by clicking a block, you can undo a connection anchor by right clicking.

Review: Connecting Condition Blocks



6. Add instruction to Action block: Turn On Inside Light.

a. Choose the Select tool.
b. Double-click the *Turn On Inside Light* block.
c. From the Instructions dialog box, click *Add*.

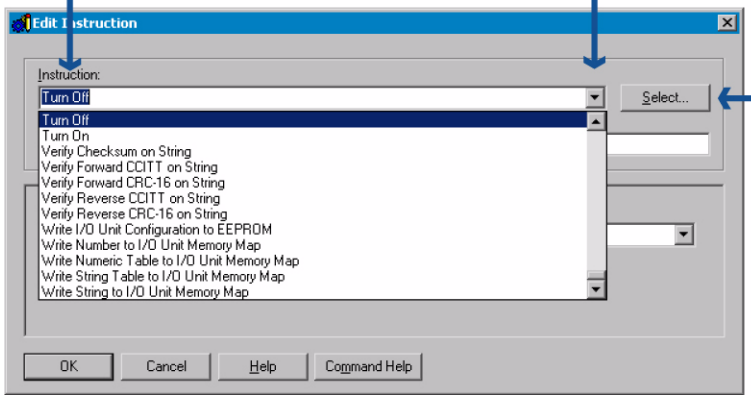


d. Select the *Turn On* instruction.

Review: Three ways to locate an Instruction:

Type the first few letters. (This finds the command Turn Off; Turn On is the next command on the list.)

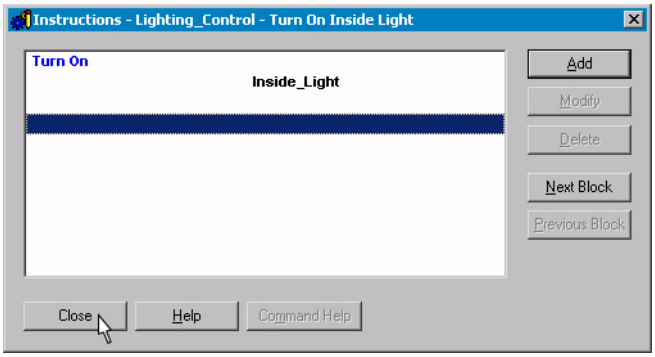
Click here to see the alphabetized list.



Click here to open a categorized list of commands.

e. Select *Digital Output* under Type.
f. Select *Inside_Light* under Name.
g. Close the Add Instruction dialog box.

Your instruction should appear as shown here:

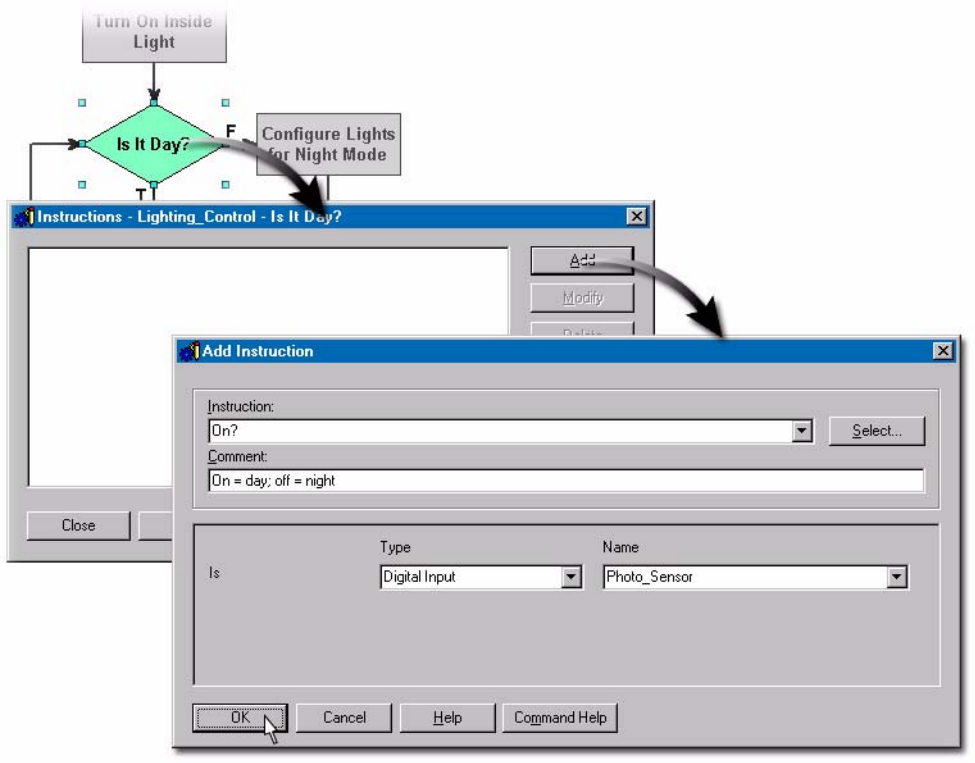


h. Close the Instructions dialog box.

7. Add Instruction to Condition block: Is It Day?

This block will check the state of the photo sensor. If the photo sensor is on, it is day; if off, it is night.

a. Double-click the *Is It Day?* block.
b. From the Instructions dialog box, click *Add.*



c. Enter the following information:

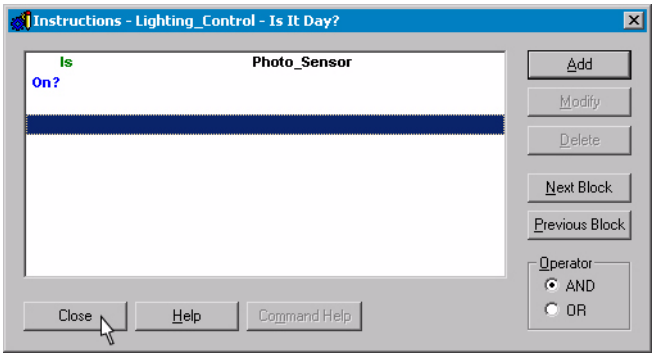
Instruction: *On?*

Comment: `On = Day; Off = night`

Type: *Digital Input*

Name: *Photo_Sensor*

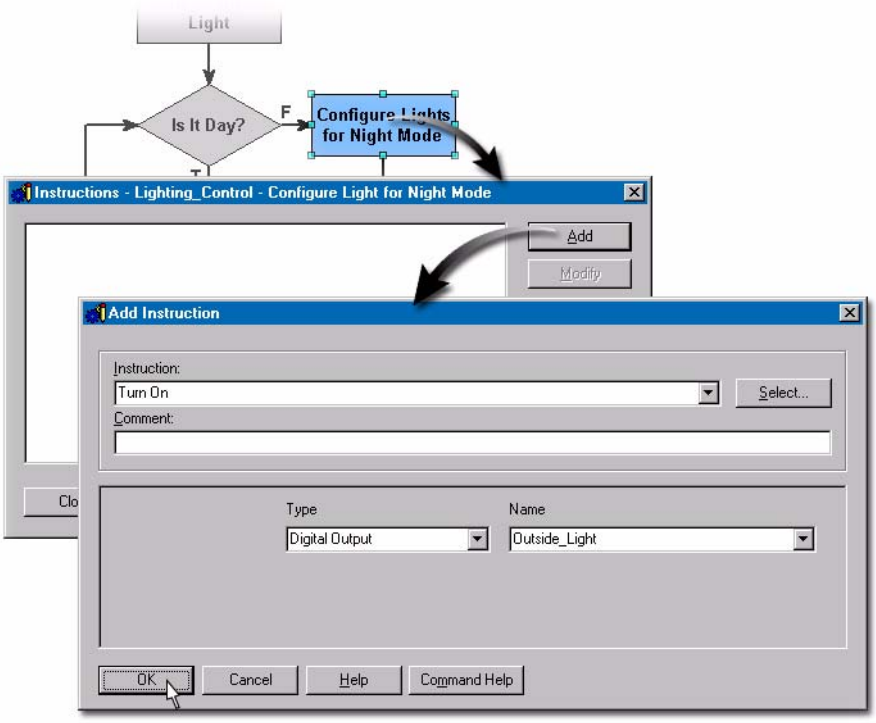
d. Click *OK* to close the Add Instruction dialog box.
Your instruction should appear as shown here:



e. Click *Close* to close the Instructions dialog box.

8. Add instruction to Action block: Configure Lights for Night Mode.

a. Double-click the *Configure Lights for Night Mode* block.
b. From the Instructions dialog box, click *Add*.

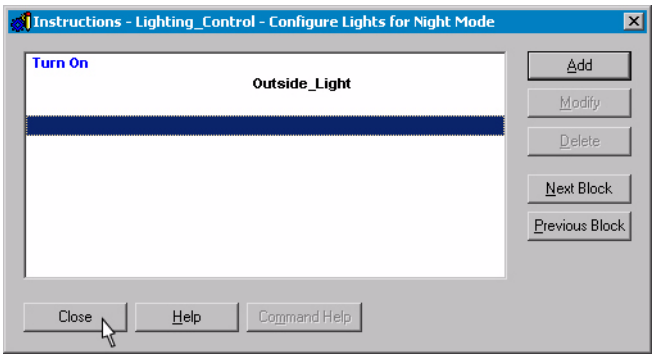


c. Enter the following information:
Instruction: *Turn On*

Type: *Digital Output*

Name: *Outside_Light*

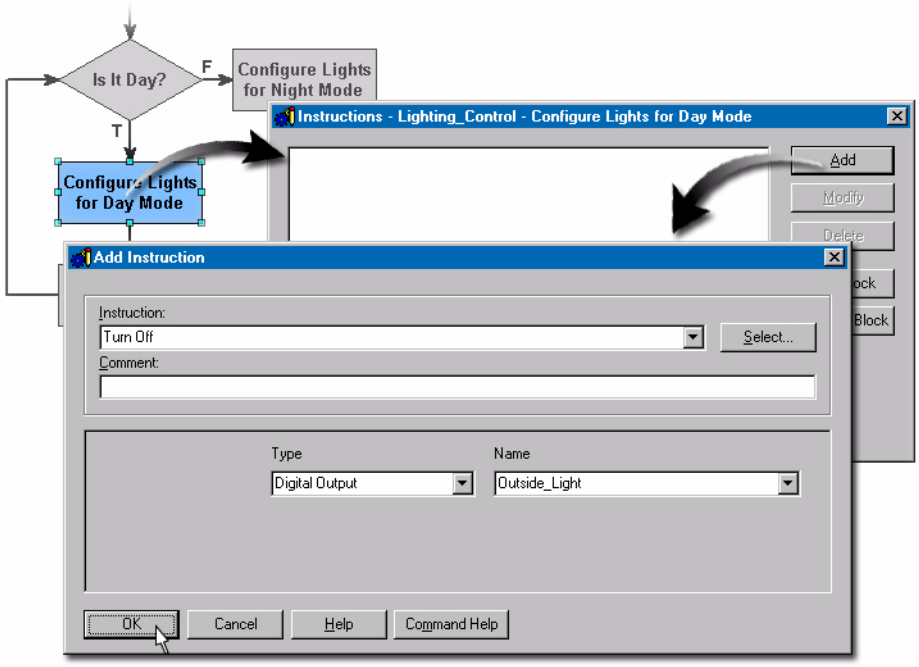
d. Click *OK* to close the Add Instruction dialog box.
Your instruction should appear as shown here:



e. Click *Close* to close the Instructions dialog box.

9. Add instruction to Action block: Configure Lights for Day Mode.

a. Double-click the *Configure Lights for Day Mode* block.
b. From the Instructions dialog box, click *Add*.



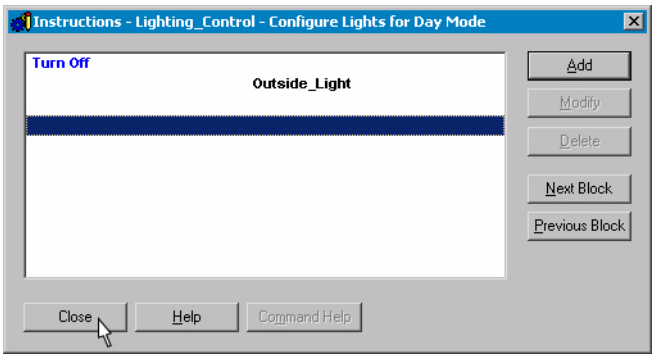
c. Enter the following information:
Instruction: *Turn Off*

Type: *Digital Output*

Name: *Outside_Light*

d. Close the Add Instruction dialog box.

Your instruction should appear as shown here:



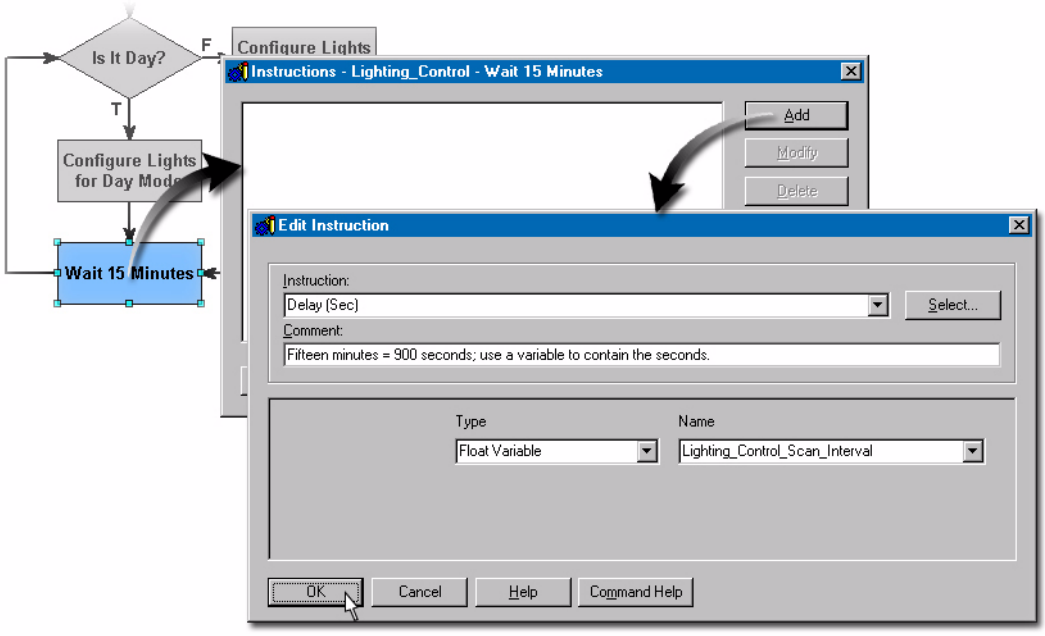
e. Close the Instructions dialog box.

10. Add instruction to Action block: Wait 15 Minutes.

In this step you create a delay. If the strategy were monitoring a real photo sensor for a gradual change from day to night, you might use a 15-minute delay. As this would be too long to observe the effect of your ioControl strategy, we will use a 2-second delay instead.

a. Double-click the *Wait 15 Minutes* block.

b. From the Instructions dialog box, click *Add*.



c. Enter the following information:

Instruction: *Delay(Sec)*

(NOTE: ioControl provides a Delay(Sec) and a Delay(mSec) command.)

Comment: `Fifteen minutes = 900 seconds; use a variable to contain the seconds.`

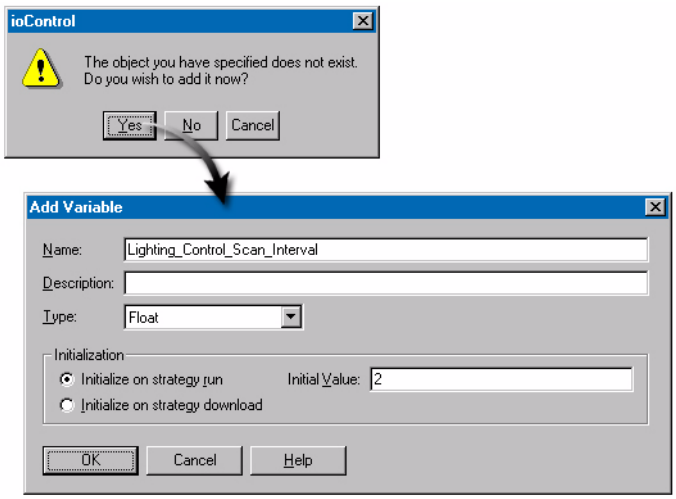
Type: *Float Variable*

Name: `Lighting Control Scan Interval`.

d. Click *OK*.
This prompts you to confirm the addition of a variable.

e. Click *Yes*.
This displays the Add Variable dialog box.

f. In the Add Variable dialog box, enter 2 for Initial Value.

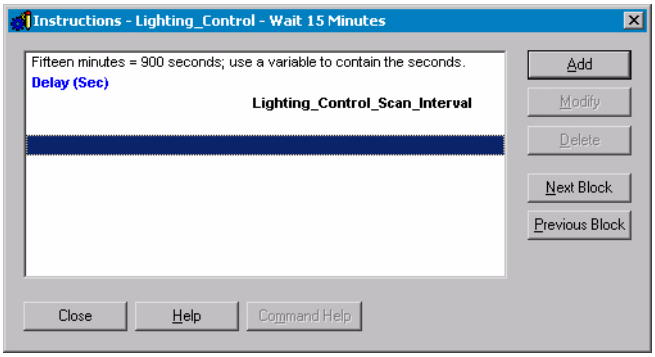


Variables vs. Literals:

When designating a delay interval, you could have used a literal instead of a variable. The literal can be changed by editing the flowchart instruction. The variable offers more flexibility: It can be changed by your strategy, by an operator using ioDisplay, or by you during Debug mode.

g. Click *OK* to add the new variable.

h. Close the Add Instruction dialog box.
Your command should appear as shown here:



i. Close the Instructions dialog box.

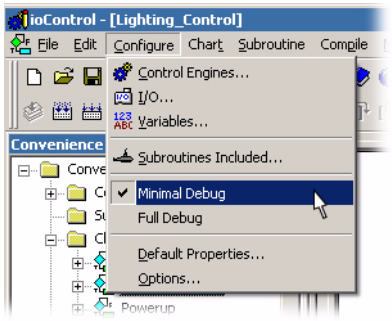
Configuring Debug Mode

You use Debug mode to monitor your strategy while it is running on the SNAP Ultimate I/O unit for the purpose of testing your strategy. ioControl has two levels of Debug, Minimal Debug and Full Debug. Minimal debug provides a smaller feature set but allows ioControl to run faster while in Debug mode. Full Debug mode offers all Debug features but takes longer to download and may slow the program significantly. In most

applications, however, you are unlikely to notice a big difference in performance between the two modes.

1. Make sure ioControl is in Minimal Debug mode.

Choose *Configure* ➞ *Minimal Debug.*



Downloading and Running the Strategy

1. Download the strategy.

a. Select *Debug* from the *Mode* menu.

The strategy will be saved, compiled, and downloaded to the control engine.

b. Acknowledge all download messages.

2. Run the strategy.

Click the *Run* button to start the strategy.

3. Test the strategy.

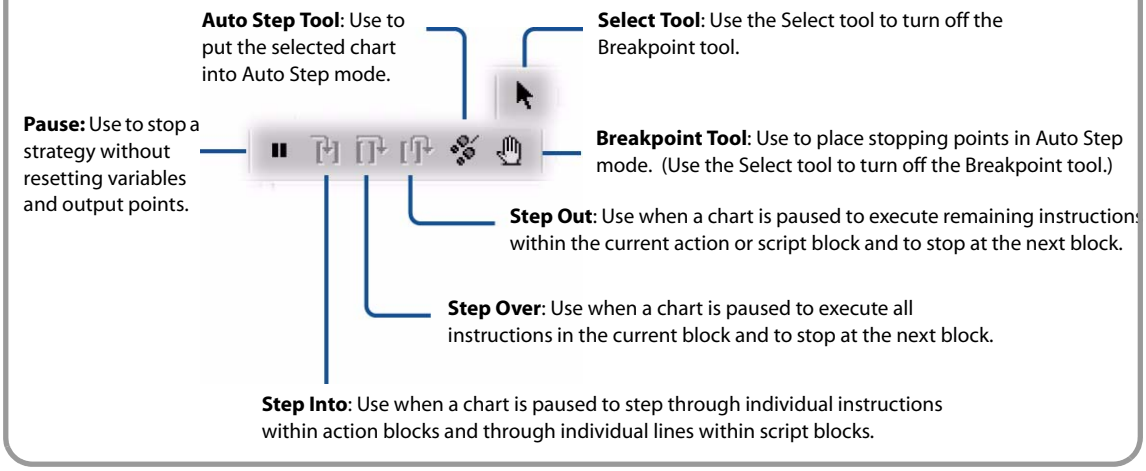
a. The Inside Light LED should be ON.
b. Turn on the Photo Sensor Day Mode switch on the load panel (toggle switch to the down position).

A photo sensor is ON when it is daylight. By turning the switch on, you are simulating daylight.

c. Observe the Outside Light turn off.
d. Turn off the Photo Sensor switch on the load panel.
e. Observe the Outside Light turn on.

Debugging Tools:

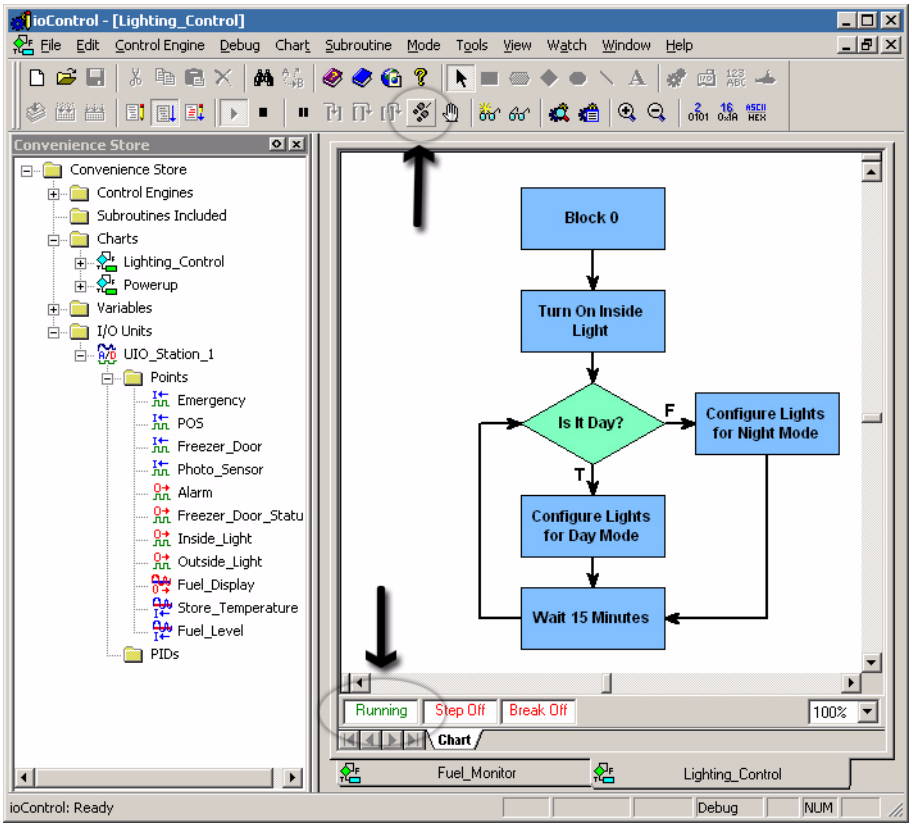
Now that your strategy performs actual logic, you can see the debugging tools in action.



Using Auto Stepping

1. Observe the strategy execution in Auto Step mode.

a. Verify that the Lighting_Control chart is running and selected (click within the chart window).



b. Click the *Auto Step Chart* button (or choose *Debug* ➞ *Auto Step Chart*).

c. Watch the execution of the Lighting_Control chart. Each block becomes cross-hatched as it is executed.

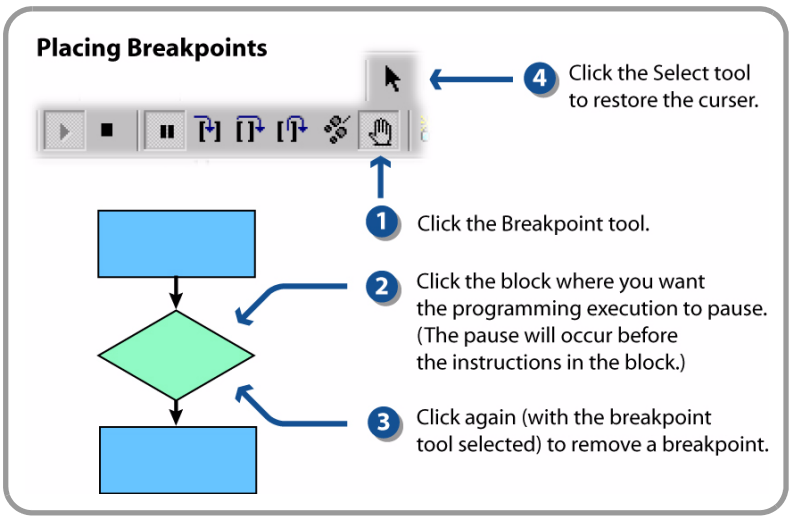
d. Turn on the Photo Sensor Day Mode switch and watch the execution of the chart change.

e. Turn off the Photo Sensor Day Mode switch.

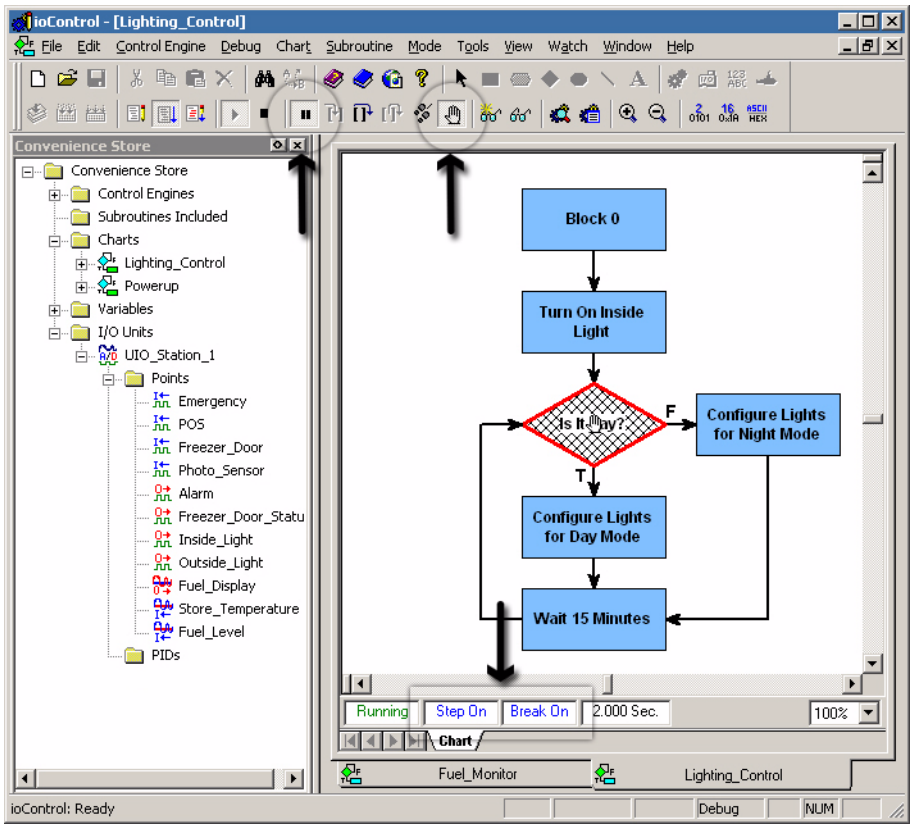
You may notice that the outside light (digital output Outside_Light) takes longer to respond to a change in the photo sensor (digital input Photo_Sensor). Auto stepping slows down the chart's execution. Once the chart is taken out of Auto Step mode, the chart will execute in real time.

2. Set a breakpoint.

a. Click the *Breakpoint* tool.
b. Click the *Is It Day?* condition block



Your strategy will run till it reaches the breakpoint in the Is It Day? condition block:



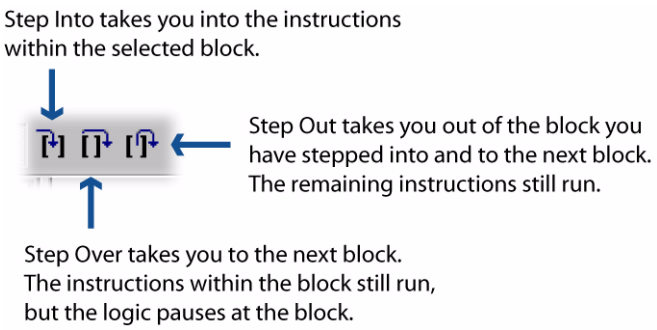
Observe that the execution of this flowchart stops at the breakpoint. The instructions in the Is It Day? block have not been executed at this point. The status of the debugging activities is shown at the bottom of the chart window. Step On and Break On indicate that the chart is paused and a breakpoint has been placed.

Also notice that the pause button is engaged. To continue Auto Step mode, remove the paused state by clicking the Pause button. To continue to the next block, click the Step Over button.

Clicking a block in the flowchart will add or remove breakpoints, as long as the Breakpoint tool remains selected. You can add and remove breakpoints with your strategy stopped or running.

Using Step Buttons

The ioControl debugger provides three tools for stepping though components of a chart: Step Into, Step Over, and Step Out.



NOTE: Step Into and Step Out are only available in Full Debug mode.

1. Enter Step On mode.

a. Click the *Pause* button.

Notice that the status changes to Step On in the bottom left of the chart window.

2. Click the *Step Over* button several times and watch the execution progress one block at a time.

3. Remove the breakpoint.

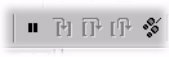
a. Select the *Breakpoint Tool*.

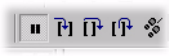
b. Click the *Is It Day?* block.

Notice that the status changes to Break Off.

c. Click the *Pause* button to continue chart execution.

Debug Tools


Pause button off (Strategy is running)


Pause button on (Strategy is stopped)

Modifying the Lighting_Control Chart

In this section, you will rewrite the Lighting_Control chart so that it uses fewer commands. You want the outside light's state to be opposite the Photo Sensor's state. This can be achieved with the *Not* command that writes the inverse of an on/off value to a point or variable.

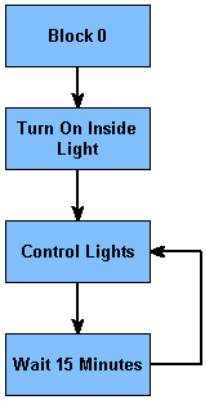
As you learn ioControl you will discover many commands that simplify your flowchart design. You will also learn that since ioControl is such a flexible, powerful programming language, there's more than one way of designing any chart.

1. Return to Configure mode.

Select *Configure* from the *Mode* menu, or click the *Configure mode* button.

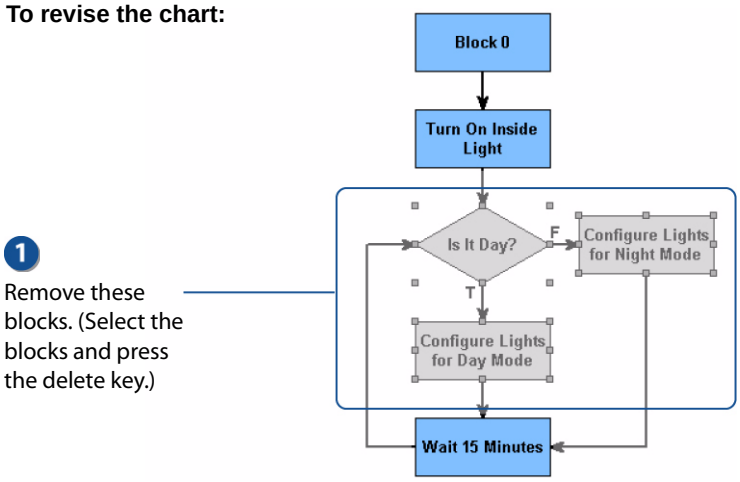
2. Modify the Lighting_Control chart.

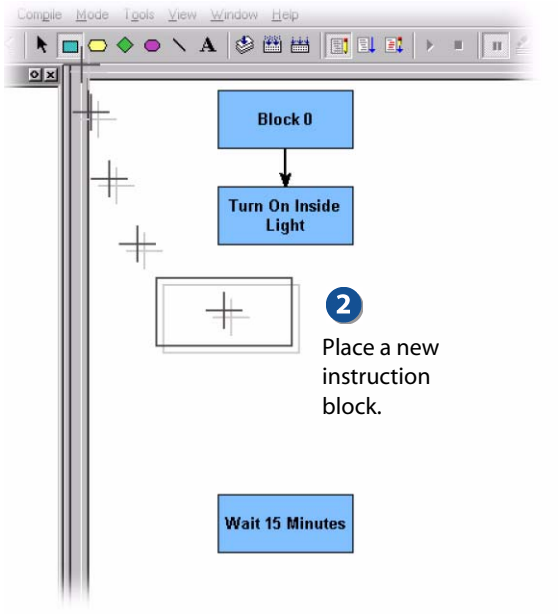
Modify your Lighting_Control chart as shown here by deleting the unneeded blocks and reconnecting the flow lines:

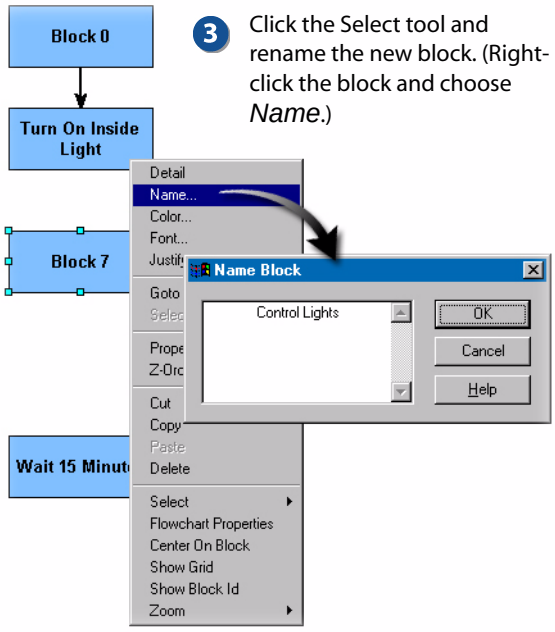


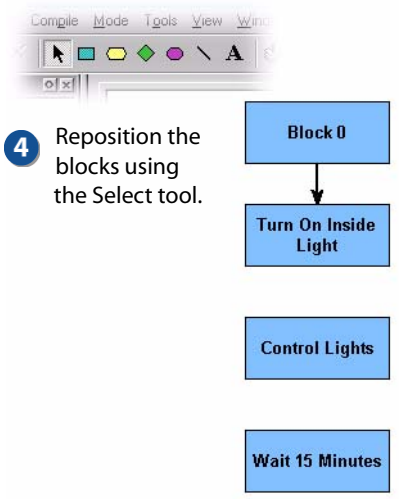
Try to revise your Lighting_Control chart on your own applying the skills you've learned. Refer to the following diagrams if you need assistance. Once you've finished this task, continue to Step 3.

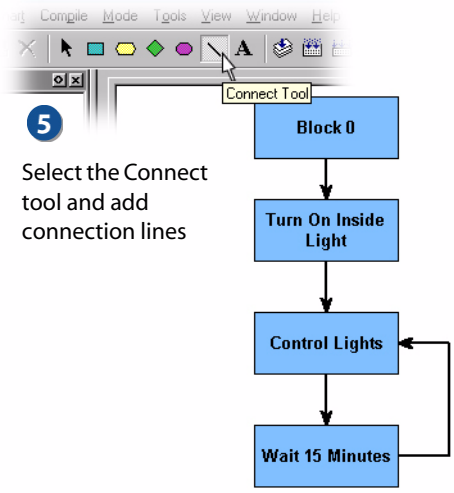
To revise the chart:







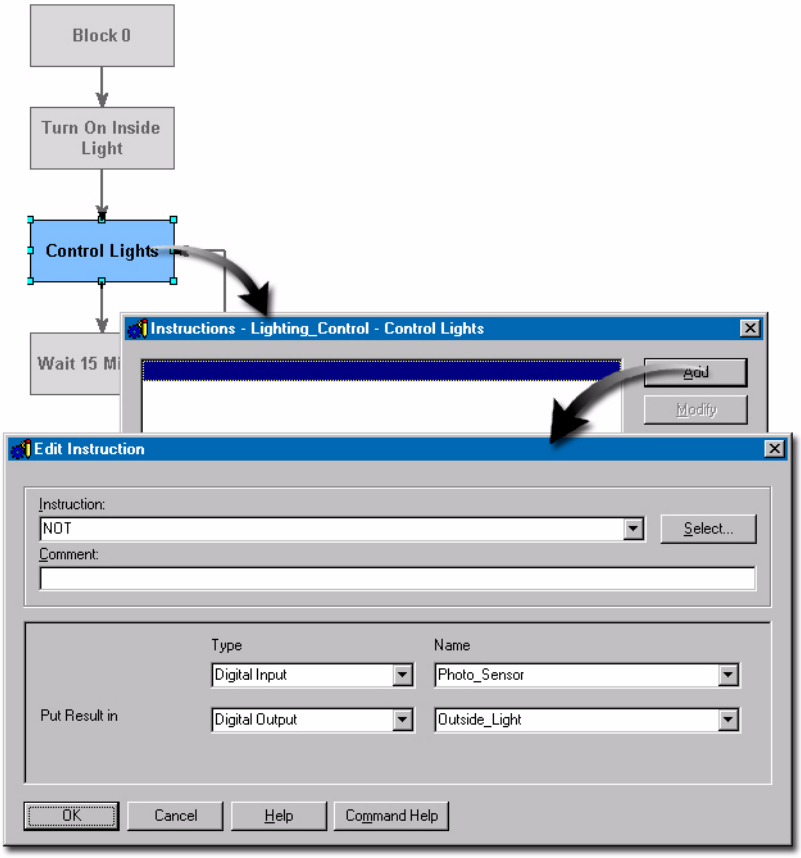




3. Add an instruction to the Control Lights action block.

a. Double-click the *Control Lights* block with the select tool to open the Instructions dialog box.

b. Click *Add* to open the Add Instruction dialog box.



c. Select *NOT* from the Instruction drop-down list.

The NOT instruction copies the inverse of information from one variable or I/O point to another.

d. Choose *Digital Input* (Type) and *Photo_Sensor* (Name) for the first parameter.

The command will read the state of the Photo_Sensor.

e. Select *Digital Output* (Type) and *Outside_Light* (Name) for the second parameter.

The command will write the inverse, or opposite, state of Photo_Sensor to Outside_Light.

f. Close all dialog boxes.

Debugging the Changes

1. Download and run the modified strategy.

a. Select *Debug* from the *Mode* menu to return to Debug mode.
b. Acknowledge all download messages.
c. Click the *Run* button.

2. Test the strategy.

a. Turn on the Photo Sensor Day Mode switch on the load panel (toggle switch in down position).

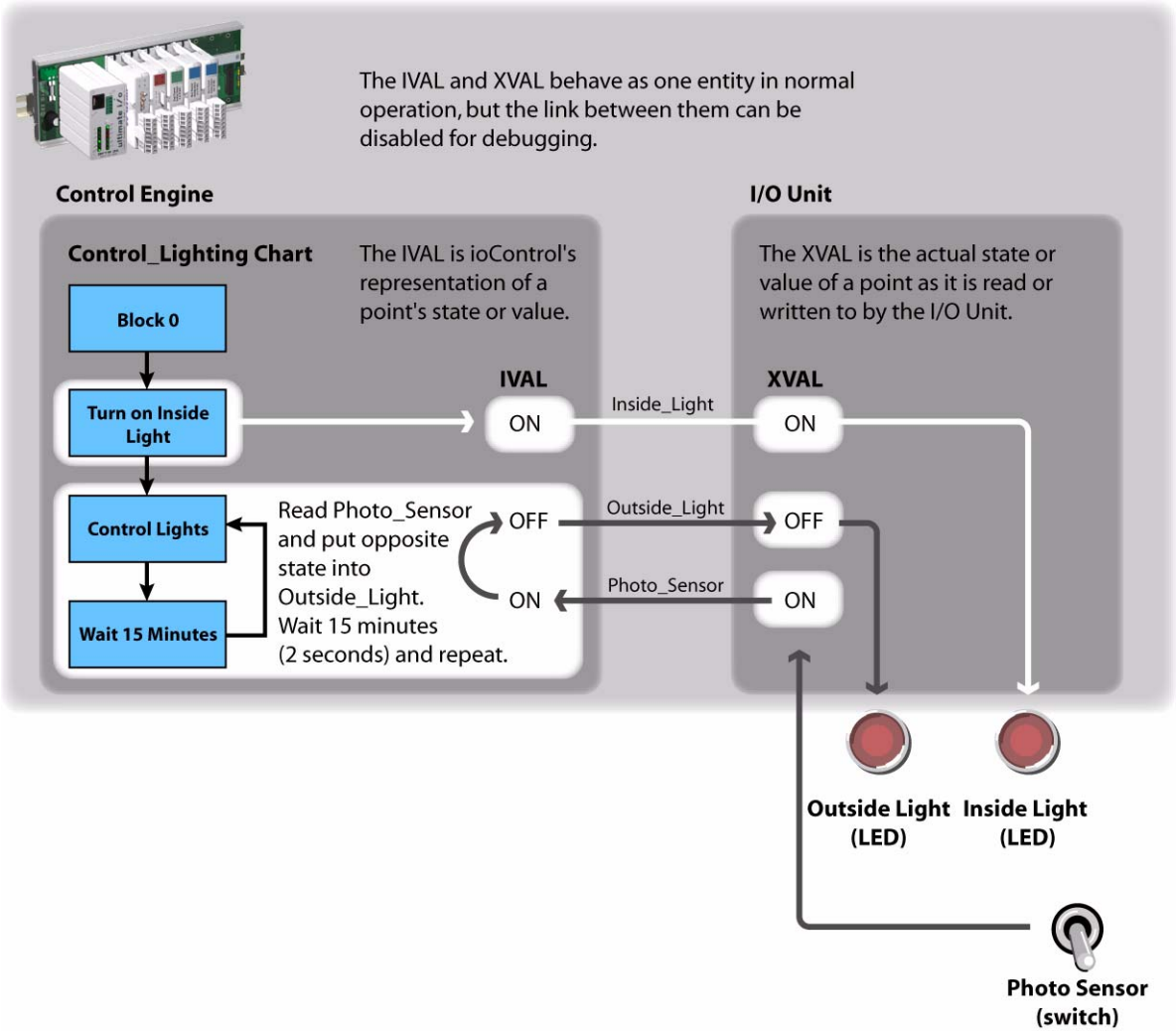
b. Observe the Outside Light turn off.
c. Turn off the Photo Sensor Day Mode on the load panel.
d. Observe the Outside Light turn on.

Forcing Outputs and Inputs in Debug Mode

Your inside light and outside light are controlled by the Control_Lighting chart. The diagram below shows what takes place between the devices on the load panel and the logic you built in the chart. There are two values for each point, an IVAL (for internal value) and an XVAL (for external value). You will now learn how to use these values for debugging.

In our scenario, we change night and day with a switch on the load panel. If you were using a real, installed photo sensor, you wouldn't be able to control night and day to test your programming. In this case, you can use Debug mode to disconnect the IVAL from the XVAL and simulate inputs to observe your program's responses.

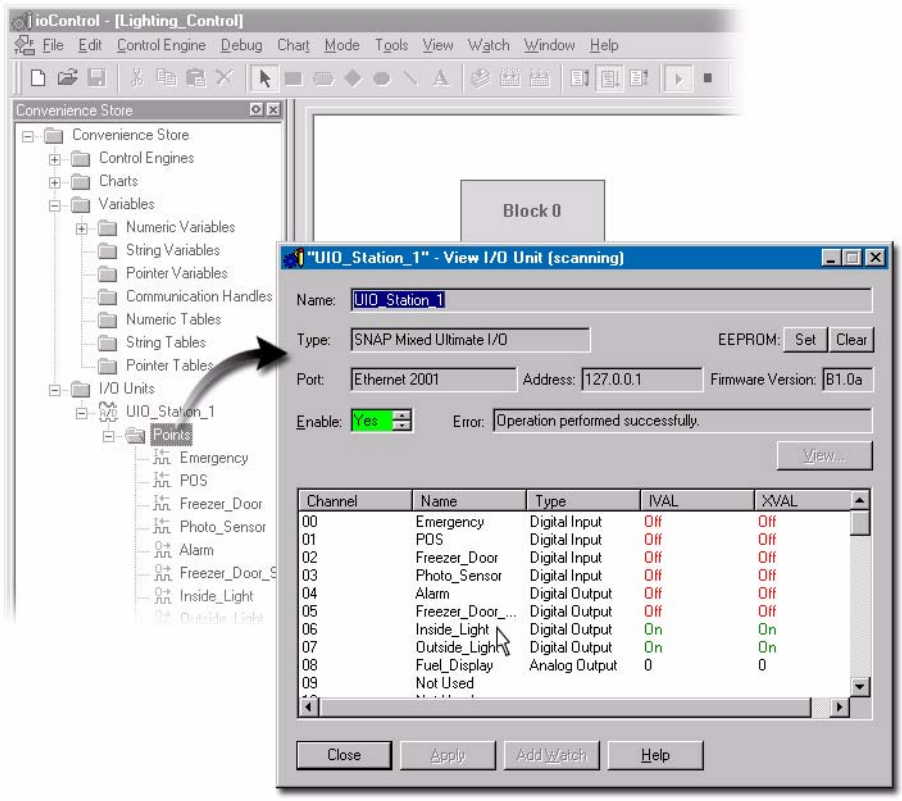
Before continuing, make sure you understand the logic of the chart.



1. Make sure you are in Debug mode.

2. View the I/O Unit.

a. Expand the I/O Units folder in the Strategy Tree.
b. Double-click *Points*.



c. On the load panel, turn the POS switch on and off.
d. Observe the IVAL and XVAL for the POS point as you turn the POS switch on and off.

   Notice that the XVAL changes but the IVAL doesn't. The IVAL is unchanged because the POS point is not referenced within the strategy.

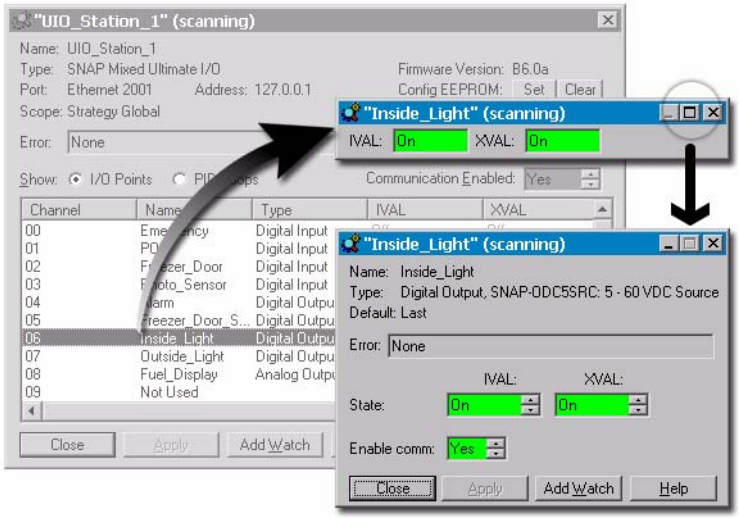
e. Observe the IVAL and XVAL for the Photo_Sensor point as you turn the Photo Sensor switch on and off.

   Both the IVAL and XVAL of the Photo_Sensor change because this point is referenced in the strategy.

3. Turn off Inside_Light.

a. Double-click *Inside_Light*.

b. In the View Digital Point dialog box, click the maximize button.



c. Click the slider arrows next to the *XVAL* field. This changes the XVAL from On to Off.

d. Click *Apply*.

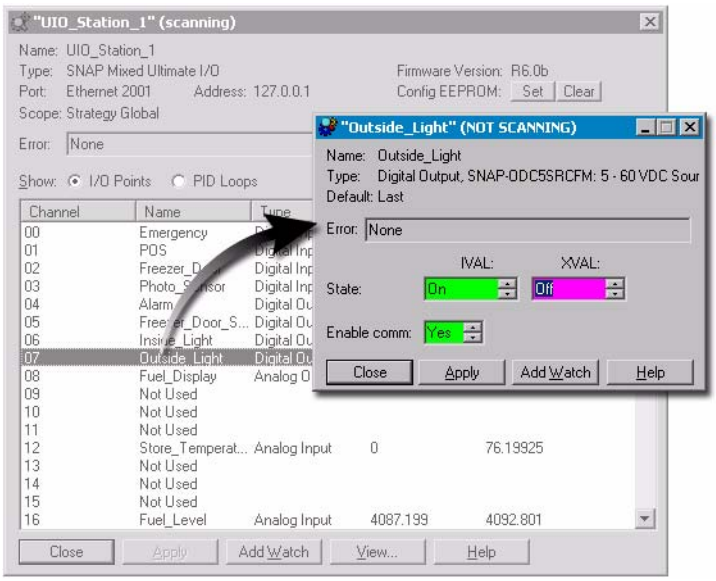
Notice that the Inside Light LED is off and stays off.

e. Close the View Digital Point dialog box.

4. Turn Outside_Light on and off.

a. Turn on the Outside Light by turning off the Photo Sensor Day Mode switch (the switch will be in its up position).

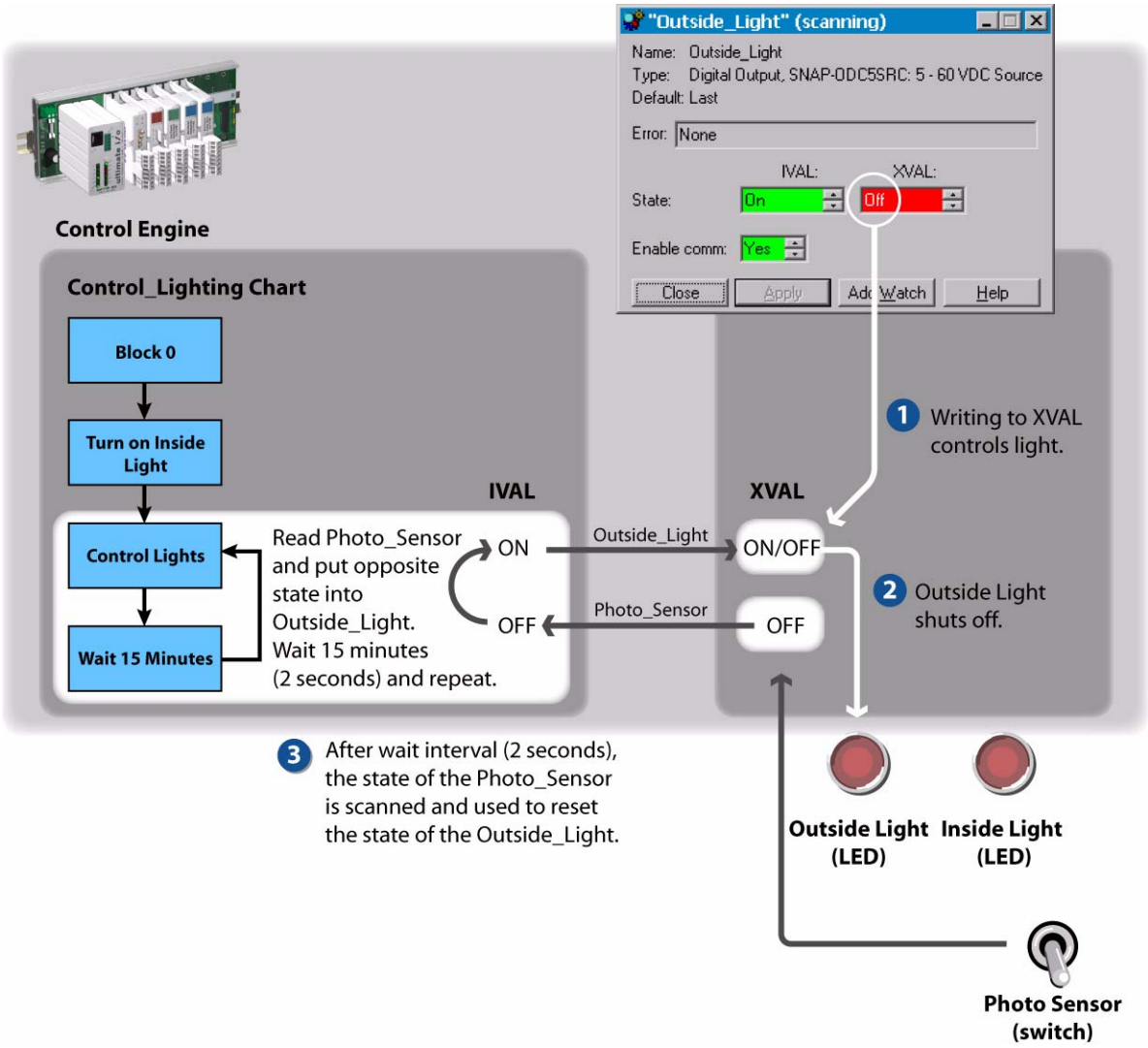
b. Open and expand the View dialog boxes for the Outside_Light.



c. Use the XVAL field in the View Digital Point dialog box to turn off the Outside Light.

d. Repeat this a few times.

Notice that the Outside Light turned on for a moment and was then turned off. This is because ioControl is programmed to turn off the Outside Light if the Photo Sensor switch is on.



5. Disable Outside_Light.

If you disable the point, you can prevent the strategy from writing to the XVAL of the Outside_Light. Once disabled, you can control the point.

a. Set the Enable Comm field to *No*, by clicking the Up/Down arrows next to the Enable field.

b. Click *Apply*.

   This breaks the link between the software and the hardware. ioControl cannot write to the XVAL of a disabled point.

c. Turn the XVAL on and off a few times. (Click the Up/Down arrows to choose *On* or *Off* and then click *Apply*.)

   Since the point is disabled, you can force the output on or off. ioControl is setting the IVAL to On, according to its logic, but this state isn't reaching the point, allowing you to turn the point on and off using the XVAL.

d. Change the Enable Comm setting back to *Yes*.
e. Close the View dialog box for Outside_Light.

NOTE: You can open more than one View dialog box for points, but in this activity, it is recommended that you keep only one of these dialog boxes open at a time to ensure that you are disabling and writing to the correct point.

6. Force input point.

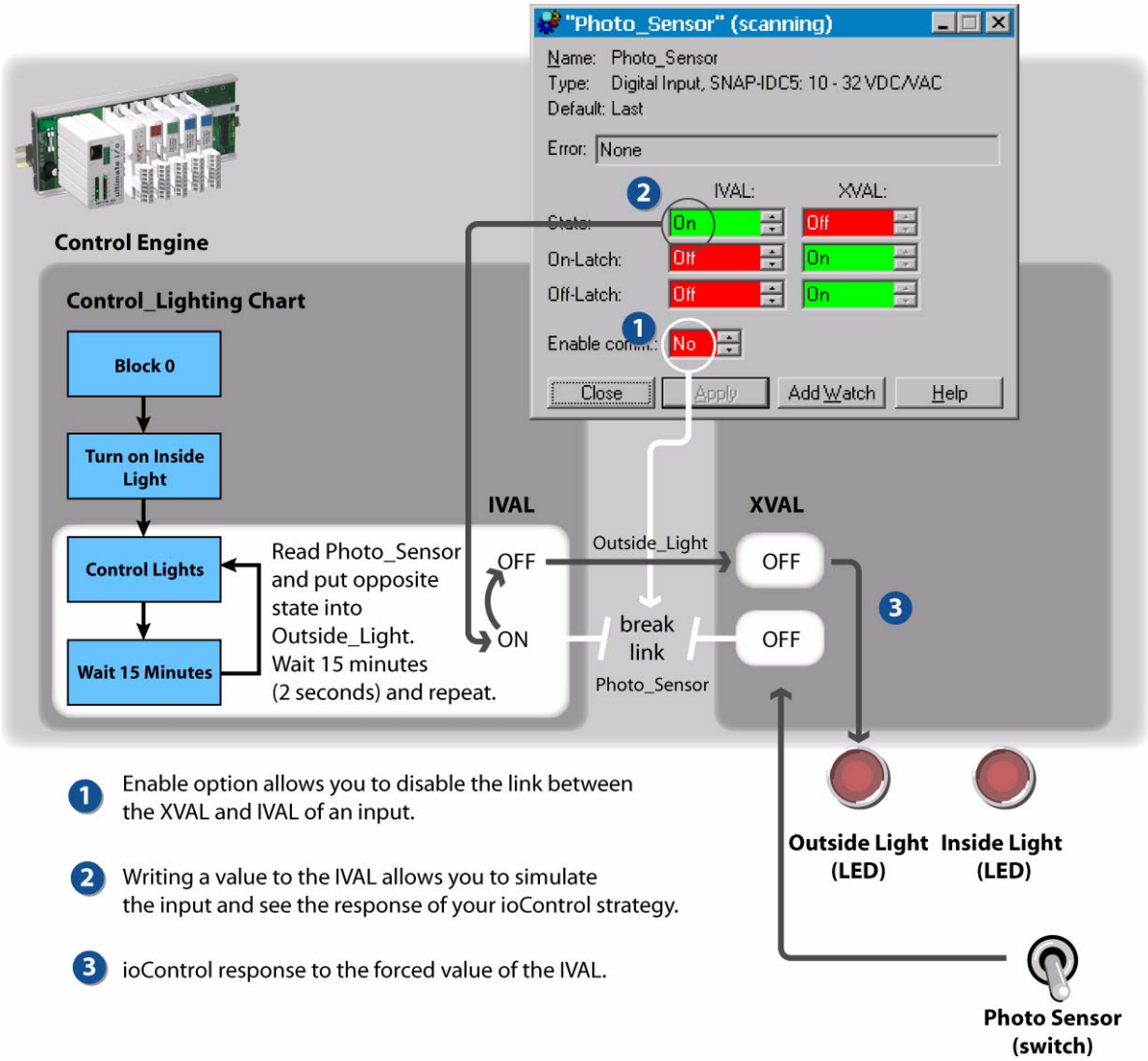
Now you will disable an input point so you can see your logic react to a simulated input.

a. Make sure your that your Outside_Light is enabled and the dialog boxes for Inside_Light and Outside_Light are closed.
b. Open and expand the View dialog box for the Photo_Sensor.
c. Disable the point *Photo_Sensor* by selecting *No* in the Enable Comm field and clicking *Apply*.

   NOTE: As the photo sensor is an input, you cannot change its XVAL from ioControl. Instead, you will change the IVAL, which is the value that ioControl reads.

d. Toggle the state of the IVAL and click *Apply*. The Outside Light should also toggle.

e. Turn the Photo Sensor switch on and off. Notice that the XVAL changes, but the IVAL doesn't.



This ability to simulate conditions is a very powerful programming tool.

f. Re-enable communication for Photo_Sensor and close its View dialog box.

Adding Control for the Freezer_Door and the Emergency Switch

Follow the next steps to modify the existing chart to complete the remaining digital control functions:

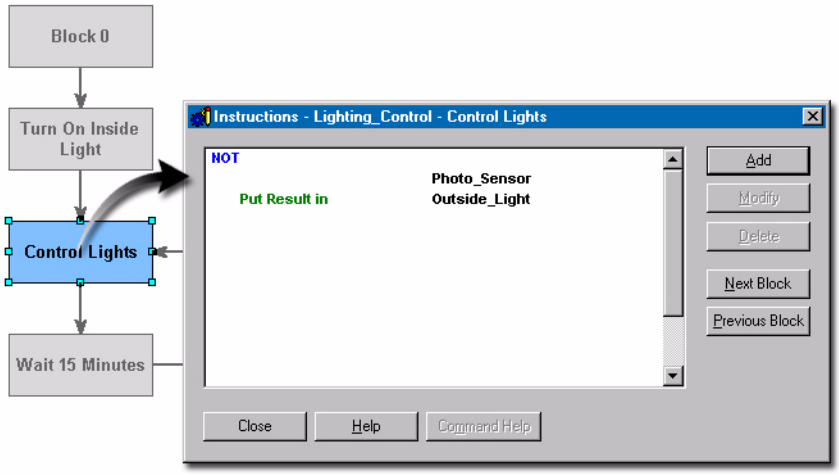
- Monitor the freezer door switch (Freezer_Door, point 2) and display the status on an LED (Freezer_Door_Status, point 5).
- Monitor the emergency switch (Emergency, point 0) and trigger the buzzer (Alarm, point 4).

1. Return to Configure mode.

Choose *Mode* ⟶ *Configure*.

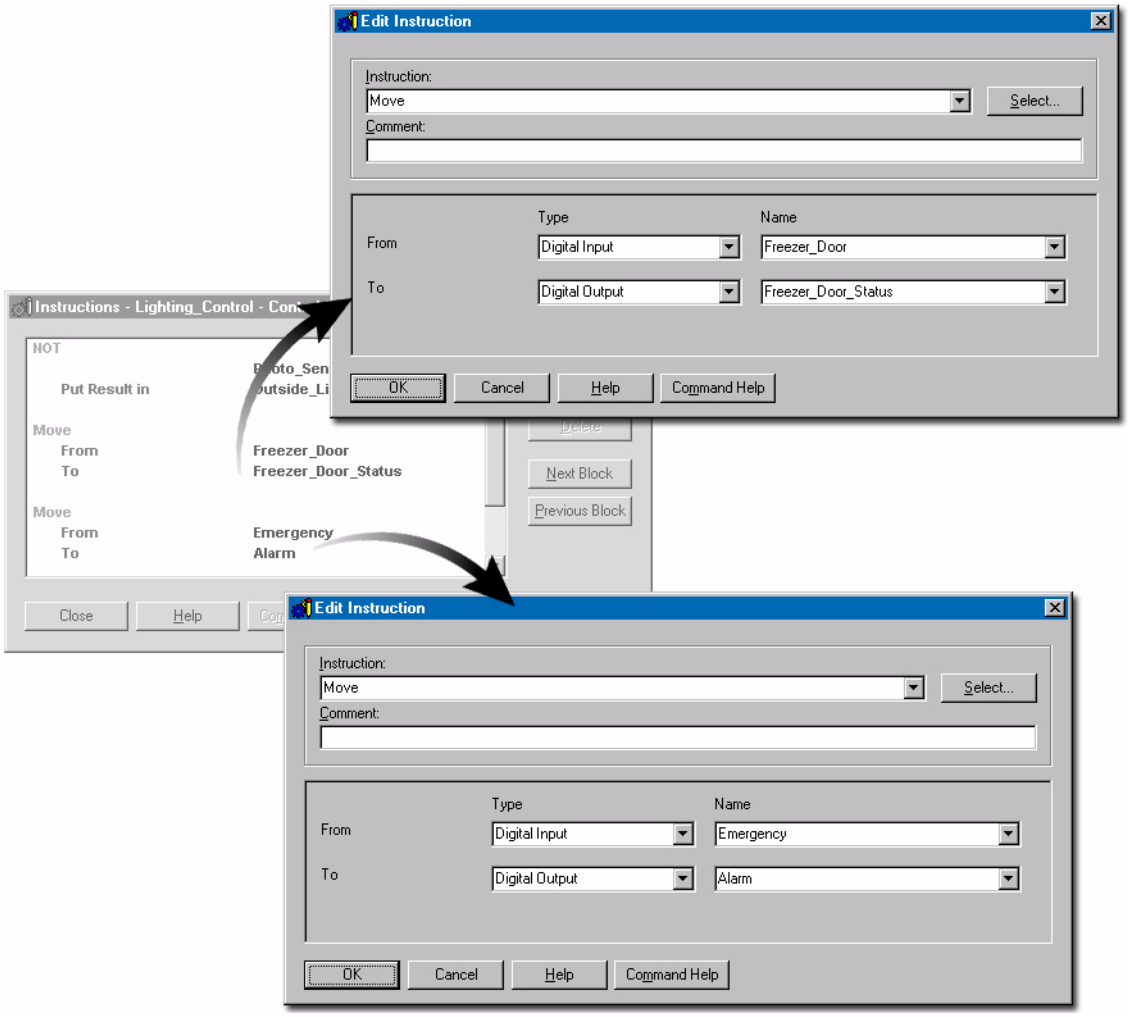
2. Modify the Lighting_Control chart.

a. Double-click the *Control Lights* Action block.



You will add two Move commands to this instruction block. The Move command is very versatile, as it allows you to read or copy information from a variety of data types.

b. Add the two Move commands shown below:



These instructions will connect the Freezer_Door input to the Freezer_Door_Status output and the Alarm input to the Emergency alarm output.

3. Download and run the modified strategy.

a. Choose *Mode* ⟶ *Debug*.
b. Acknowledge the download messages.
c. Choose *Debug* ⟶ *Run*.

4. Test the strategy.

a. Press and hold the *Emergency* button to trigger the alarm.

   The Alarm button must be pressed while the program loops into the *Control Lights* block.

b. Once the alarm is on, release the Alarm switch.

   Within two seconds the alarm will be turned off.

c. Turn on the Freezer Door switch to trigger the Freezer Door Status light.

5. Return to Configure mode.

Choose *Configure* under the *Mode* menu.

You will further modify this strategy in Lesson 5, but will not make any more changes to the Lighting_Control chart. Below is a follow-up activity to try on your own to practice the skills you learned in this lesson.

FOLLOW-UP

The strategy you just built monitors three input points and uses the state of these points to control three output points. The strategy achieves this in a loop that checks and writes to points and then pauses before repeating. In this lesson, we controlled the pause through a variable with the value of 2. A 15-minute delay was adequate for automating the outdoor lights, but is much too long for the Emergency and Freezer door. A simple way to solve this is to put the functions in a different chart according to the length of pause the function needs.

This solution is using ioControl's multitasking. You can create many independent charts and up to seven can run simultaneously.

How would you make this change to your strategy? One method is described below.

Creating separate charts:

a. Create a new chart named *Frequent_Monitor*.
b. Copy the flowchart blocks *Wait 15 Minutes* and *Control Lights...* from Lighting_Control and paste them into the Frequent_Monitor chart.

In Frequent_Monitor chart:

c. Rename the *Control Lights...* block to *Monitor Emergency and Freezer Door*.
d. Edit the *Wait* instruction so that the delay is 2 seconds: for example, use a string literal set to 2, or create a new variable with an initial value of 2.
e. Connect *Block* 0 to the *Wait* block and connect the *Wait* block to *Monitor Emergency and Freezer Door.*
f. Open the details of *Monitor Emergency and Freezer Door* and delete the Control Lights command (the NOT command). Make no changes to the Move commands.

In Lighting_Control chart:

g. View the details of the *Control Lighting* block and remove the Move commands for the Freezer_Door and the Alarm.

Variables

h. Set the initial value of Scan Interval to 900 (15 minutes).

In Powerup chart:

i. Add a new command to the Start Chart blocks, starting the Frequent_Monitor chart.

Analog Control

Skills

Analog points

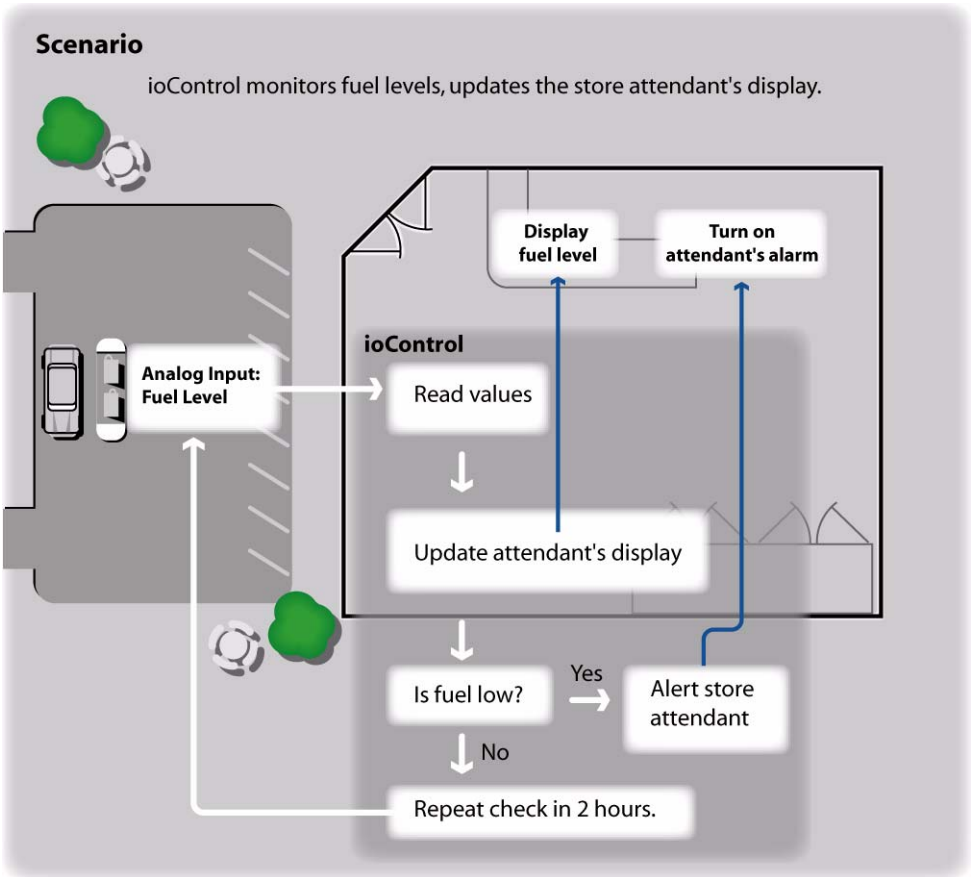
- Referencing analog I/O
- Using variable and literal floats

Debugging

- Changing variable values in real time
- Using Watch Windows

Scenario

In this lesson, you will use SNAP Ultimate I/O to control analog values and to sound an alarm when the fuel tank level is low.



Our digital controls are taken care of for the moment. The Lighting_Control chart provides the following control: your outside light responds to the daylight; your inside light is on when the system is running; your freezer door notifies the store attendant when it's left open; and an alarm sounds when a switch is pressed. These activities continue, untouched by the new programming you'll add. The chart you create in this lesson will do the following:

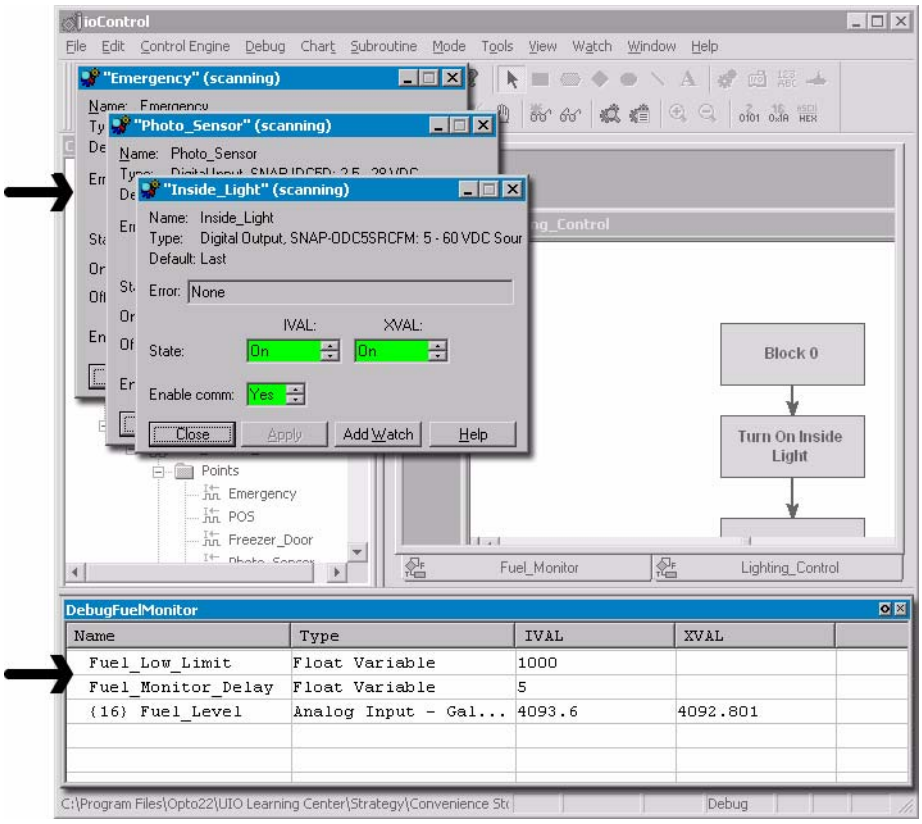
- Read the level of the fuel tank (simulated by the Tank Level potentiometer).
- Display the current level of the tank on the store attendant's monitor (simulated by the meter).
- Evaluate the tank level.
- Respond to a low level with an brief alarm.
- Repeat at a reasonable interval (using a delay command).

CONCEPTS

Watch Windows

In this lesson you will use a Watch Window as a debugging tool. Watch Windows allow you to preserve a list of variables that you need to examine together.

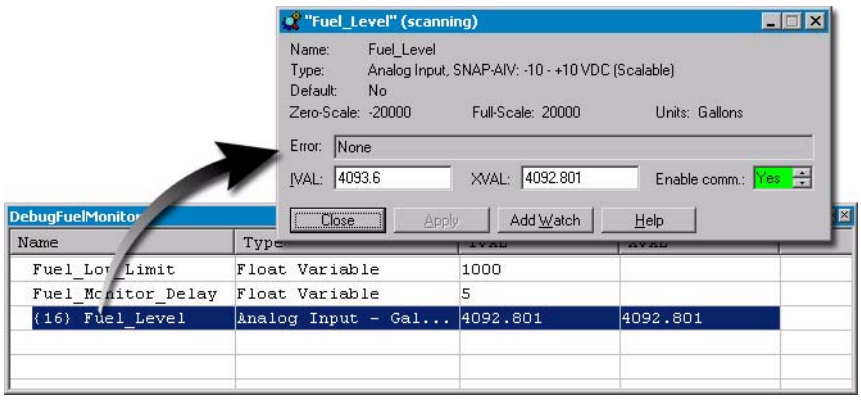
Scanning dialog boxes are closed when you exit Debug mode. Each item must be opened and closed individually.



Watch Windows can contain any data from the strategy and are saved under a unique file name.

A Watch Window is saved as a separate file, and if left open during Debug mode, it is automatically opened when you return to Debug mode.

Watch Windows are easy to create and also allow you to open scanning dialog boxes to write to variables or points.



ACTIVITY

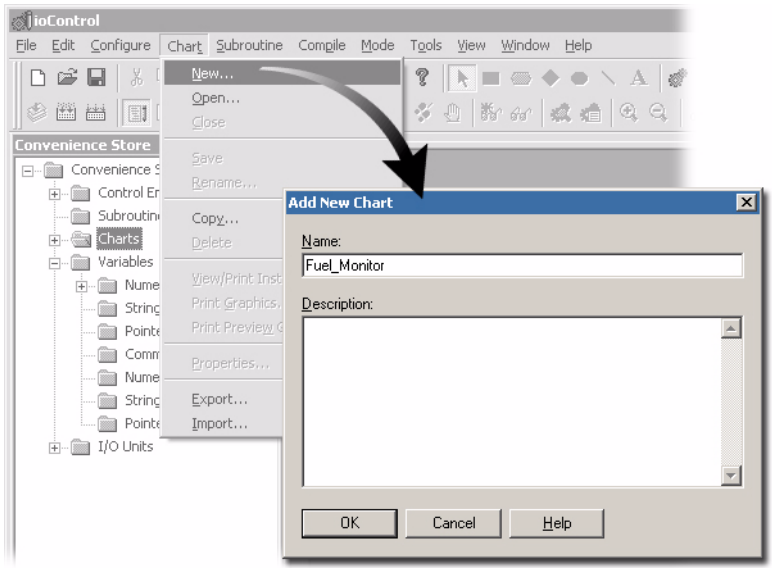
Creating a Fuel Monitor Flowchart

1. Make sure you are in Configure mode.

Choose menu command *Mode* ⟶ *Configure.*

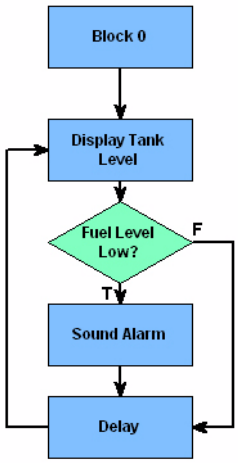
2. Create a new flowchart.

a. Select *Chart* ⟶ *New.*
b. Name the new chart `Fuel_Monitor`.



c. Click *OK* to close the Add New Chart dialog box.

3. **Create the flowchart as shown here:**



Review: Chart Tools

Selection tool.
Use to manipulate flowchart objects.

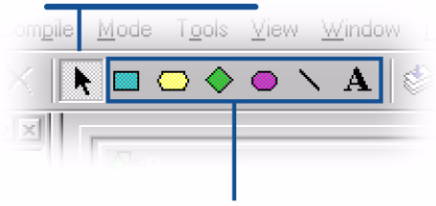
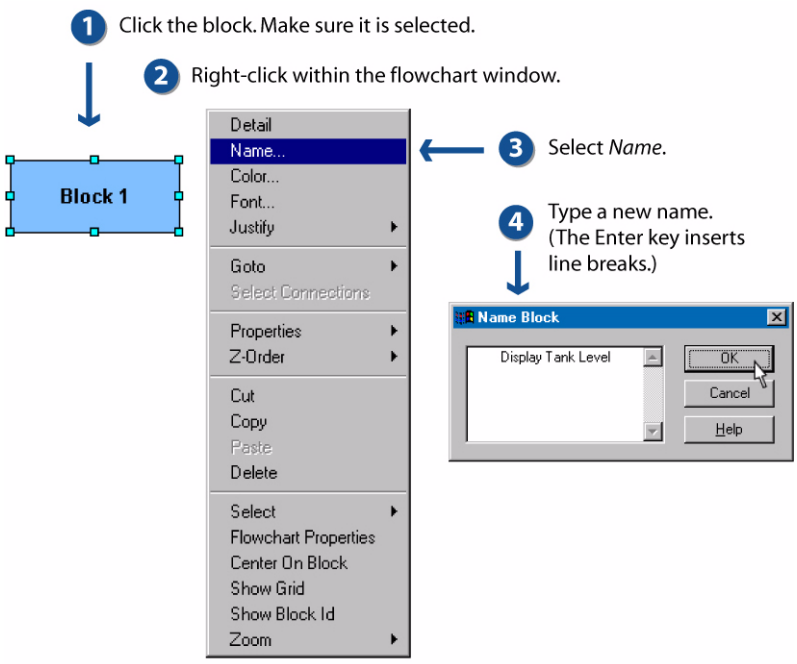
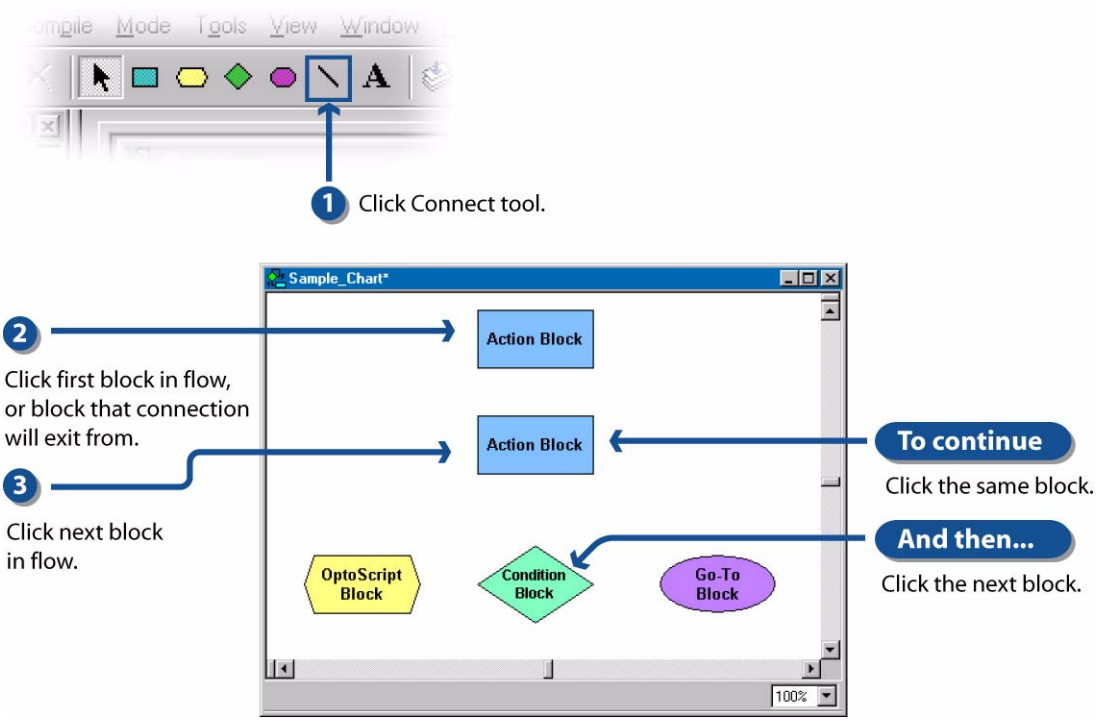


Chart objects: Once selected, each of these tools places its corresponding object until you choose the Select tool or a different chart object tool.

Review: Renaming a Block



Review: Connecting blocks

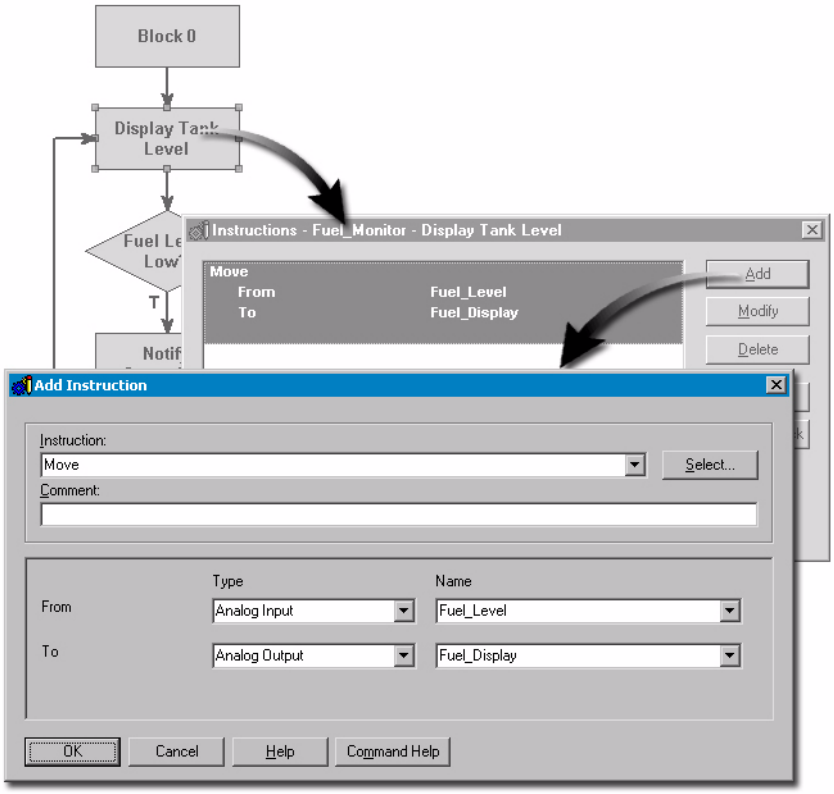


Reading and Displaying Fuel Levels

1. Create instructions for Display Tank Level.

This block will move the value of analog input point Fuel_Level to analog output point Fuel_Display.

Create the instruction shown below:



Summary of Steps:

1. Create a Move command:
   a. Double-click *Display Tank Level*.
   b. Click *Add*.
   c. In the Instruction field, select the *Move* command.
   d. In the From field, select *Analog Input–Fuel_Level*.
   e. In the To field, select *Analog Output–Fuel_Display*.
2. Close the Edit Instruction dialog box.
3. Close the Instructions dialog box.

NOTE: You could also move the value of the I/O to a variable in your own strategies. This would be advantageous when you want to minimize the frequency with which the control engine scans the I/O. By storing an I/O state in a variable, you can reference the variable, rather than the I/O, each time you need the value. This minimizes the number of times that the I/O point is polled and cuts down on communication time.

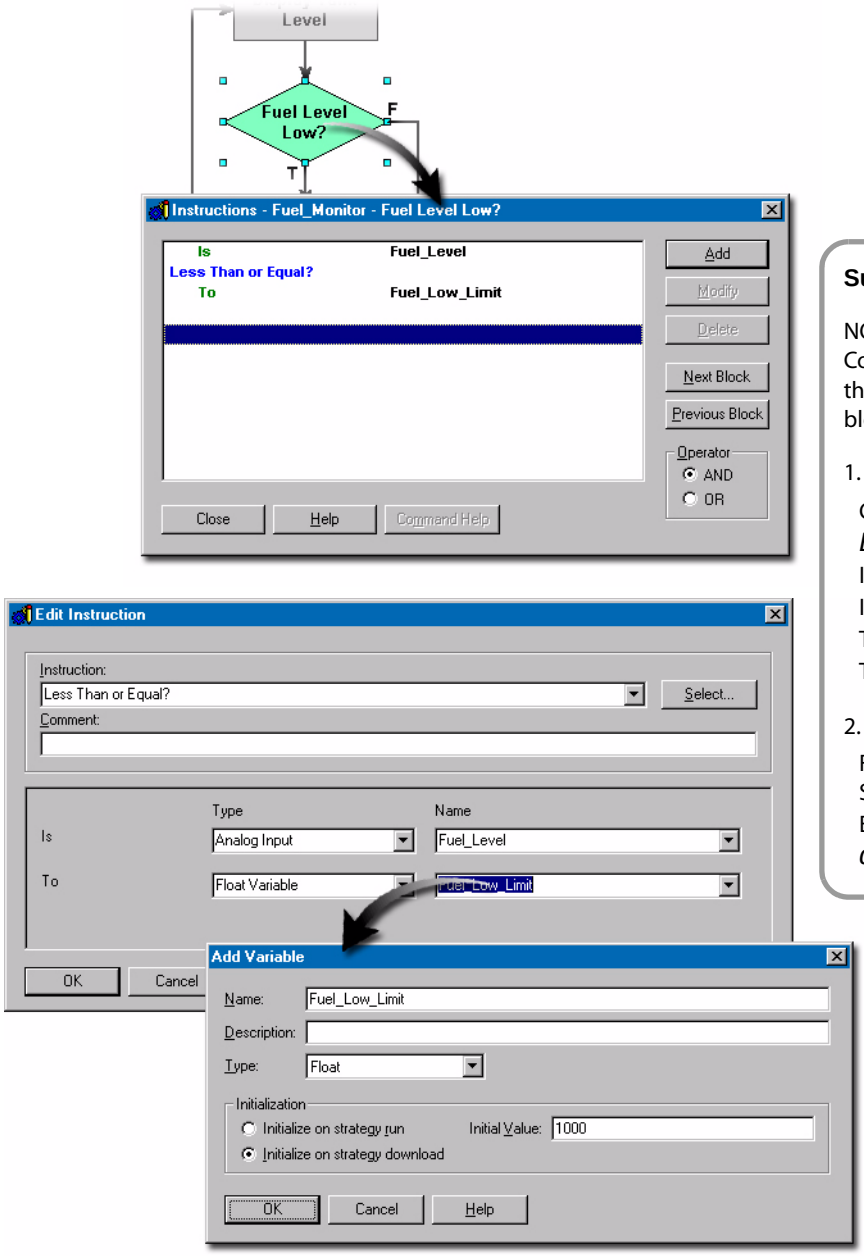
Evaluating the Tank Level

1. Create instruction for Fuel Level Low?

This block compares Fuel_Level with the variable Fuel_Low_Limit. If Fuel_Level is less than the Fuel_Low_Limit, the block exits true.

a. Double-click *Fuel Level Low?*

b. Enter the instruction shown below.



Summary of Steps:

NOTE: The instruction set for a Condition block is different from the instruction set for an Action block.

1. Create a Condition command:

   Command: *Less Than or Equal?*
   Is–Type: *Analog Input*
   Is–Name: *Fuel_Level*
   To–Type: *Float Variable*
   To–Name: *Fuel_Low_Limit*

2. Create new float variable:

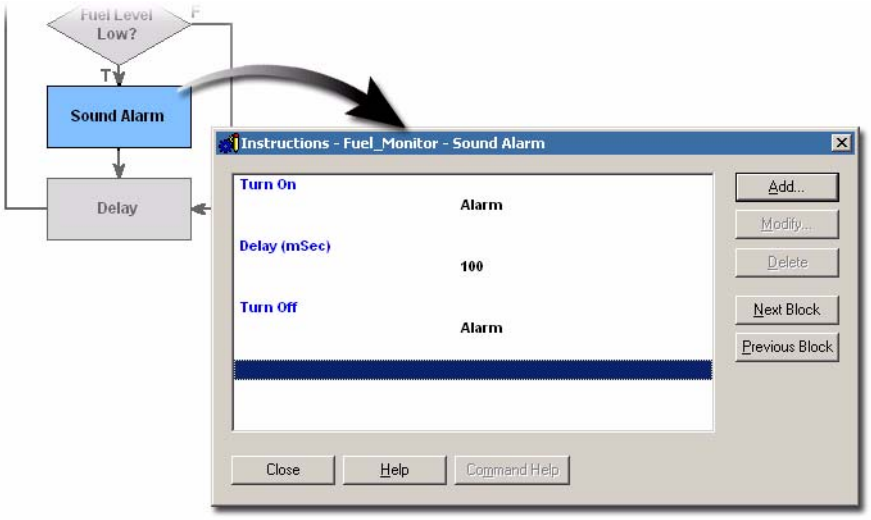
   Fuel_Low_Limit
   Set initial value to 1000.
   Enable *Initialize on strategy download.*

Sounding an Alarm

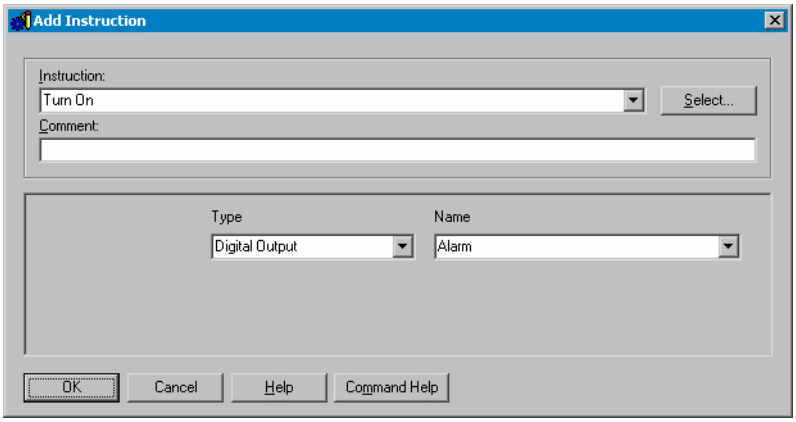
1. Create instructions for Sound Alarm.

You will create three instructions for this block. The first instruction turns on the alarm. The second instruction inserts a short delay (so that the alarm is not immediately turned off). The third instruction turns off the alarm.

Use the skills you've learned so far to create these instructions in the Sound Alarm block:



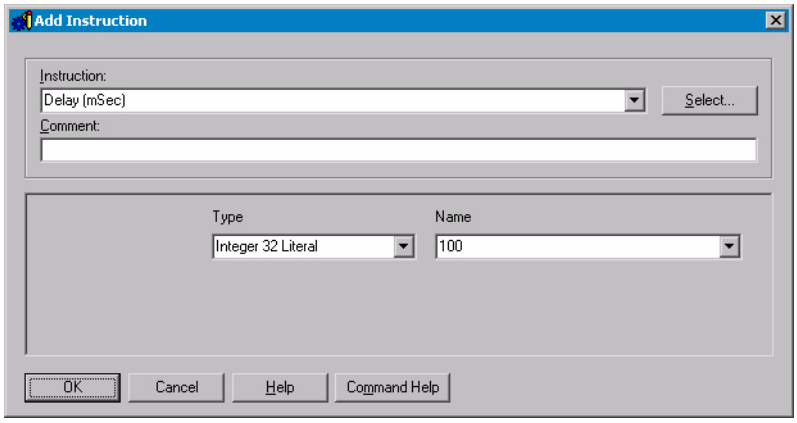
Summary of Steps: Turn on the alarm



1. In the Instructions dialog box, click *Add* to open the Add Instruction dialog box.

2. In the Edit Instruction dialog box:
   Select the instruction *Turn On*.
   Select *Digital Output–Alarm*.
   Click *OK*.

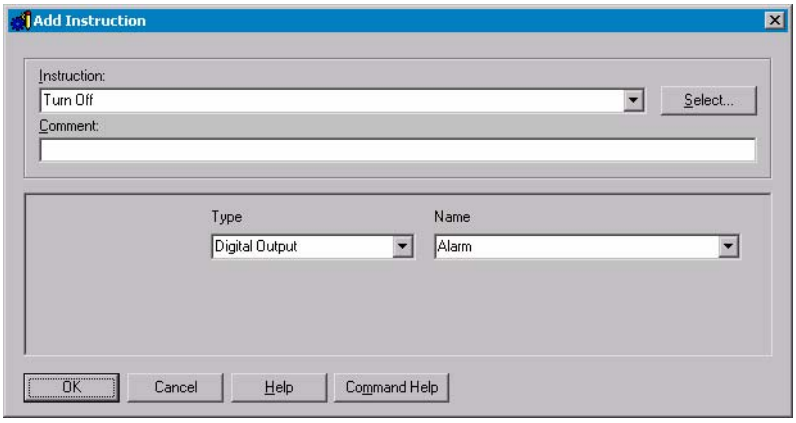
Summary of Steps: Delay



1. In the Instructions dialog box, click *Add* to open the Add Instruction dialog box.

2. In the Edit Instruction dialog box:
   Select the instruction *Delay(mSec)*.
   NOTE: Make sure you choose *Delay(mSec)* and not *Delay(Sec)*.
   Select Integer 32 Literal and enter the value 100.
   Click *OK*.

Summary of Steps: Turn off the alarm



1. In the Instructions dialog box, click *Add* to open the Add Instruction dialog box.

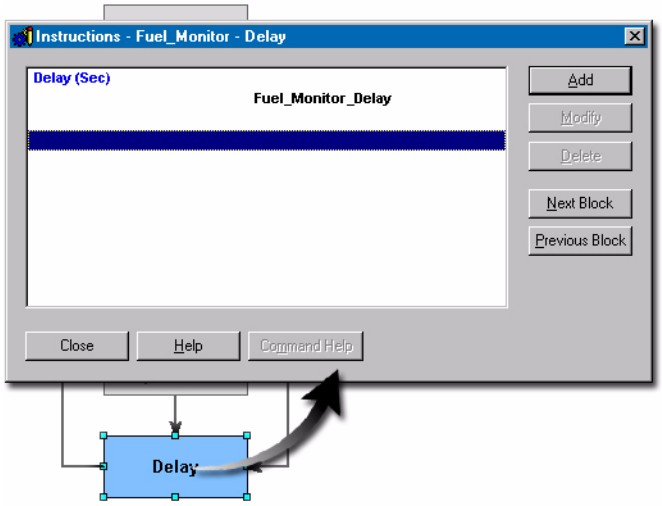
2. In the Edit Instruction dialog box:
   Select the instruction *Turn On*.
   Select *Digital Output–Alarm*.
   Click *OK*.

Repeating at a Reasonable Interval

1. Create a delay instruction.

a. Double-click the *Delay* action block.

b. Enter the instruction shown below.

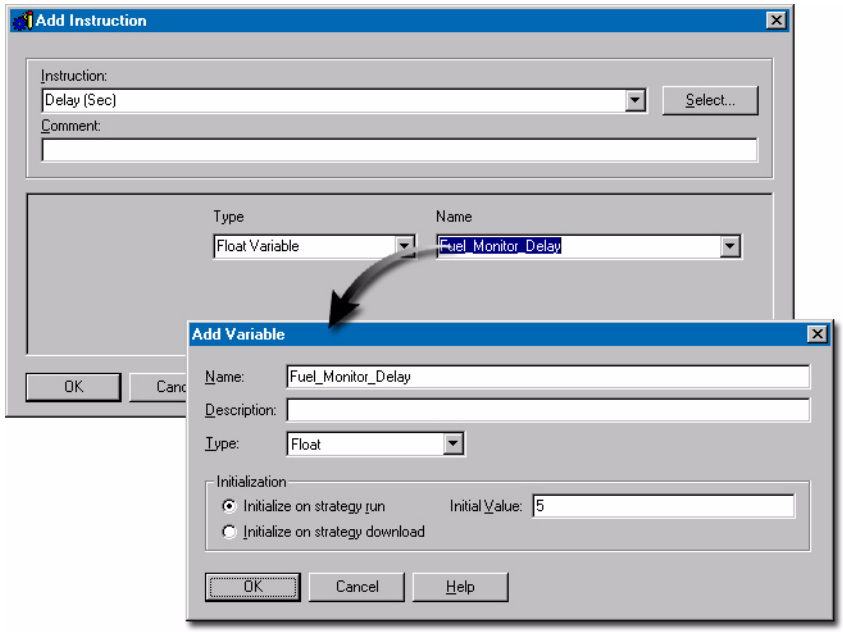


Summary of Steps

Create a Delay instruction:

1. Select the instruction *Delay(Sec)*.
   NOTE: Make sure you choose *Delay(Sec)* and not *Delay(mSec)*.
2. Create a new float variable named Fuel_Monitor_Delay.
3. Assign an initial value of 5.

(In reality, you'd want to use a longer delay so you wouldn't get an alarm every 5 seconds telling you that the fuel is low.)



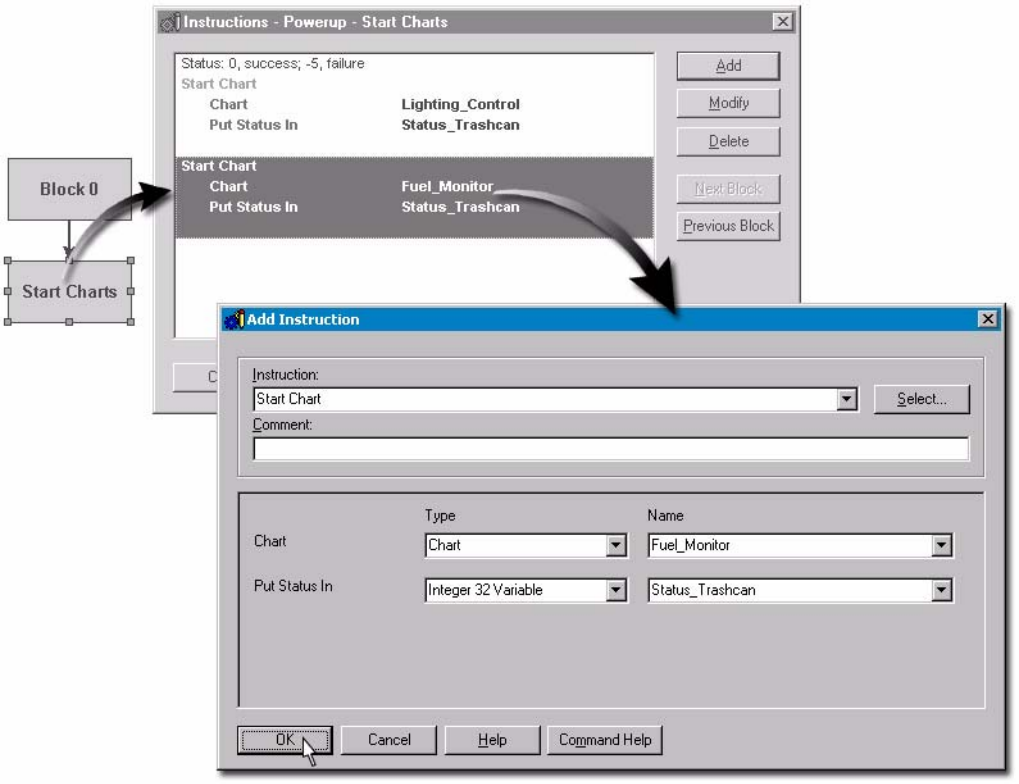
Starting the Fuel Monitor Chart

Remember: Only the Powerup chart starts automatically upon running the strategy. You need to put a command in the Powerup chart to start the Fuel Monitor Chart.

1. Modify the Powerup chart.

a. If the Powerup chart is open, click the *Powerup* tab at the bottom of the ioControl window. If it is not open, double-click *Powerup* in the Charts folder in the Strategy Tree.

b. Add the Start Chart instruction for the Fuel_Monitor chart as shown here.



c. Close the Add Instruction dialog box.
d. Close the Start Charts Instructions dialog box.

2. Save your strategy.

Downloading and Testing Your Strategy

Your revised strategy now turns on a brief alarm when fuel levels reach 1000 gallons or lower.

NOTE: By default, ioControl does not automatically run the strategy on powerup. You can change this setting in the Download–Flash Memory options, available through *File* ➞ *Strategy Options*.

1. On the Learning Center load panel, set the Tank Level potentiometer to midrange.

2. Download and run your strategy.

3. View the Fuel Monitor chart.

If the Fuel_Monitor chart is open, click the *Fuel_Monitor* tab at the bottom of the ioControl window. If it is not open, double-click *Fuel_Monitor* in the Charts folder in the Strategy Tree.

4. Turn on Auto Stepping if you would like to view the execution of the flowchart.

a. Click the Fuel_Monitor chart.
b. Choose *Debug* ➞ *Auto Step Chart*.

5. Test the strategy.

a. Turn the Fuel potentiometer knob counterclockwise.
b. Watch the needle on the Fuel Display meter track the value (with a five-second delay).

   When the value of the tank level is less than the low tank limit of 1000 (represented by 1 on the meter scale), the alarm sounds.

c. Turn the fuel level potentiometer to below 1000.

As long as the tank level is below 1000 gallons, the Ultimate I/O brain will sound an alarm every five seconds.

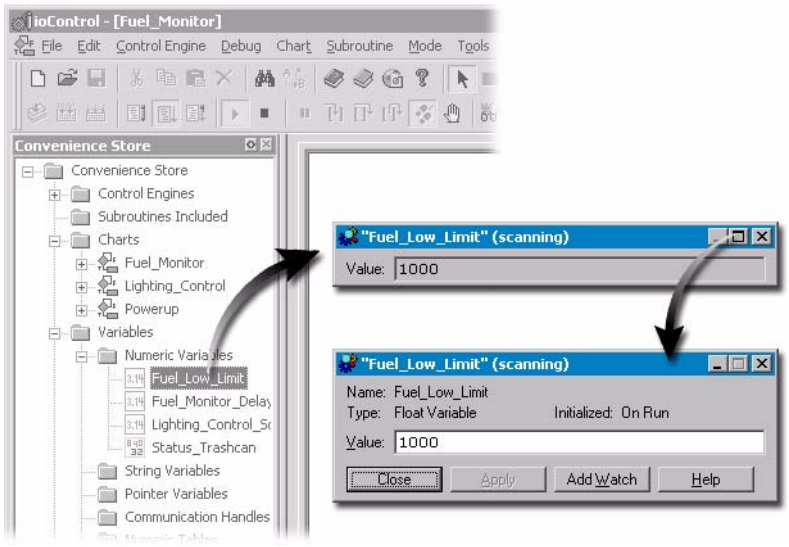
Changing the Value of Fuel_Low_Limit in Real Time

The low level value used to determine when to sound the alarm is contained in the variable Fuel_Low_Limit. You can change the variable in Debug mode.

1. Open the View Variable dialog box for Fuel_Low_Limit.

a. Make sure that your strategy is still running and ioControl is in Debug mode.
b. In the Strategy Tree, expand the *Variables* folder.
c. Expand the *Numerical Variables* folder.
d. Double-click *Fuel_Low_Limit*.
e. Expand the *View Variable* dialog box.
f. Type `2000` in the *Value* field.
g. Click *Apply*.
h. Verify that the alarm sounds any time fuel level is below 2000.

NOTE: The variable Fuel_Low_Limit is initiated on strategy download. If you stop and restart the strategy, the value will remain 2000. However, if you download the strategy, the value will be initialized to 1000.



Leave the Fuel_Low_Limit View Variable dialog box open; you will use it to create a Watch Window.

Using Watch Windows

A Watch Window allows you to monitor numerous strategy elements simultaneously and save the window for reuse. You can monitor I/O units, digital and analog points, event/reactions, variables, and even charts in a Watch Window.

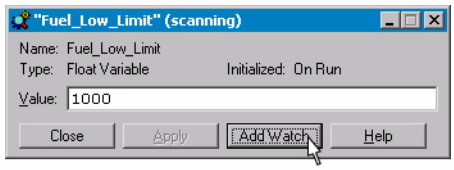
Unlike Inspection Windows (such as, View Variable windows), Watch Windows

- can be created the way you want
- can be docked in a position most convenient for you
- are saved with your strategy.

You cannot change strategy elements in a Watch Window, but you can open the Inspect dialog box from the Watch Window to change the element.

1. Add a variable to the Watch Window.

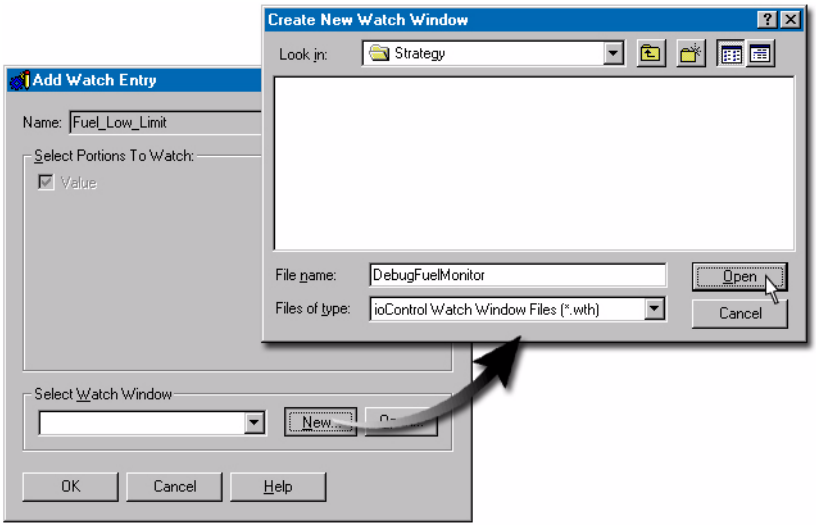
a. In the Fuel_Low_Limit inspect dialog, click *Add Watch*.



b. Click *New* to create a new Watch Window.

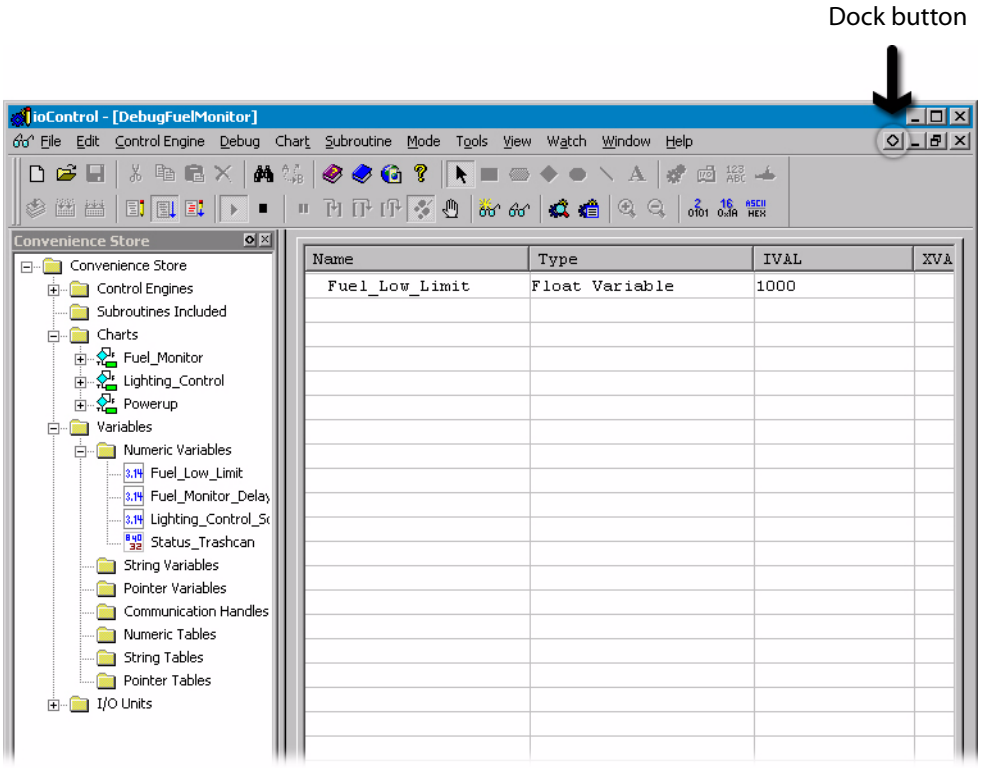
c. Navigate, if necessary, to the **C:\Program Files\Opto 22\UIO Learning Center\Strategy** directory.

d. Type `DebugFuelMonitor` in the *File name* field.



e. Click *Open*.
f. Click *OK* to close the Add Watch Entry dialog box.
g. Click *Close* to close the "Fuel_Low_Limit" View Variable dialog box.

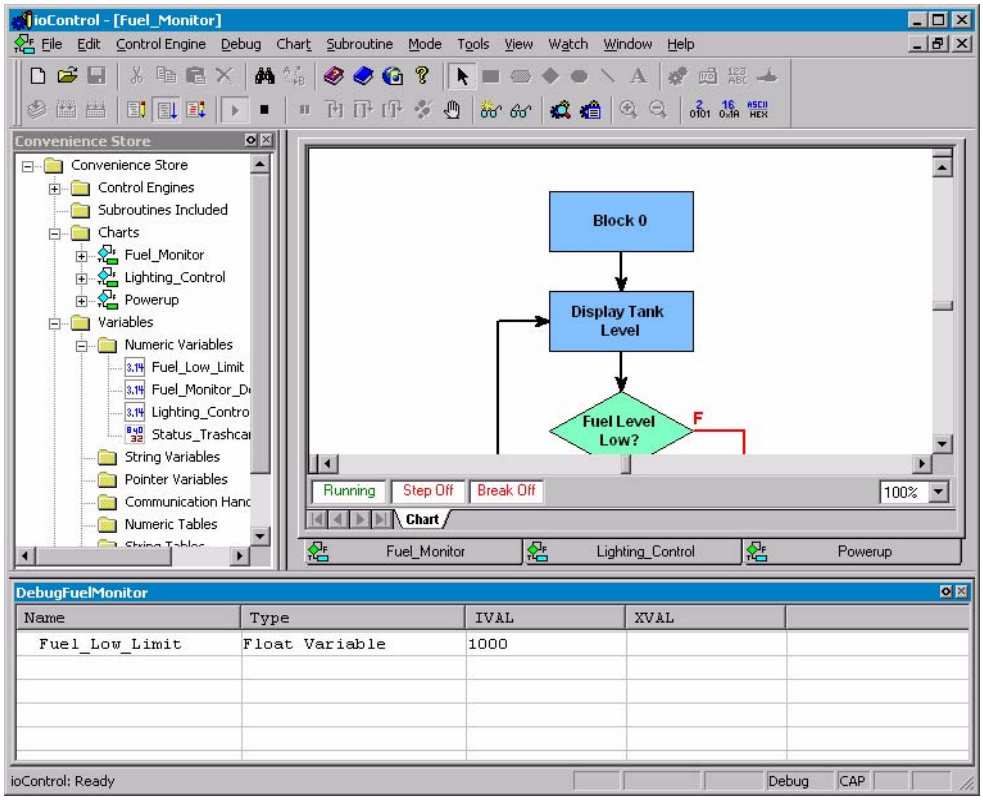
The new Watch Window appears.



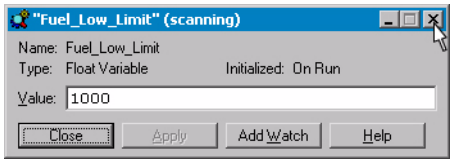
2. Arrange your workspace.

a. Notice the Dock, Minimize, and Maximize buttons in the top right-hand corner of the Watch Window.
b. Click the Dock button ( ).

The Watch Window is now docked at the bottom of your workspace as shown below.



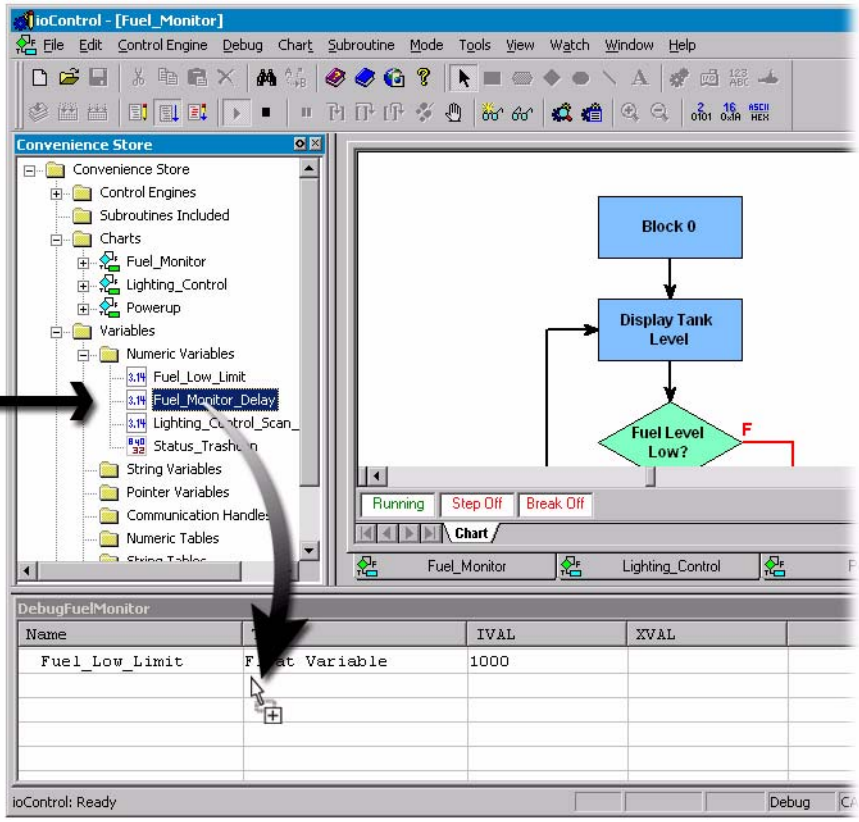
c. Close the View Variable dialog box.



3. Add variables to the Watch Window.

Click and drag *Fuel_Monitor_Delay* to the Watch Window.

Click and drag
Fuel_Monitor_Delay



You can add additional items to the watch window by clicking any tag in the Strategy Tree and dragging it to the Watch Window.

4. Add Fuel Level to the Watch Window.

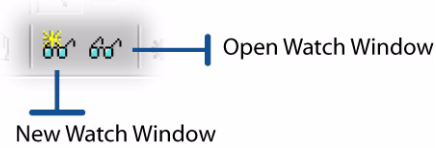
a. Open the I/O Units folder in the Strategy Tree.
b. Expand folders until the I/O points are shown.
c. Click and drag *Fuel_Level* to the Watch Window.

5. Experiment with the Watch Windows.

Here are a few things you can do:

- Close the Watch Window. To reopen, click the *Open Watch Window* button or select *Watch* ⟶ *Debug* from the top menu.
- Delete an item. Just highlight a tag name, right-click, and select *Delete*.
- Move an item up or down in the list. Highlight a tag name, right-click, and select *Move Up* or *Move Down* (or click and drag a tag name).
- Open an inspect dialog by double-clicking a tag name.

Watch Window buttons:



6. Close the Watch Window.

Your Watch Window is saved. You can create additional Watch Windows.

7. Return to Configure mode.

In Lesson 6 you will revise your Fuel_Monitor chart. Leave ioControl and your strategy open if you plan to continue to the next lesson.

OPTOSCRIPT

SKILLS

OptoScript

- Creating OptoScript blocks
- Using variables
- Using analog points
- Using commands in OptoScript
- Using the OptoScript reference library
- Writing conditional statements
- Using OptoScript's test compiler

Debugging OptoScript

- Setting Full Debug option
- Stepping line by line

SCENARIO

The chart you created in the previous exercise displays the fuel level on your meter and checks the level against the minimum defined by the variable Fuel_Low_Limit. To demonstrate OptoScript, we're going to rewrite the chart using one OptoScript block.

CONCEPTS

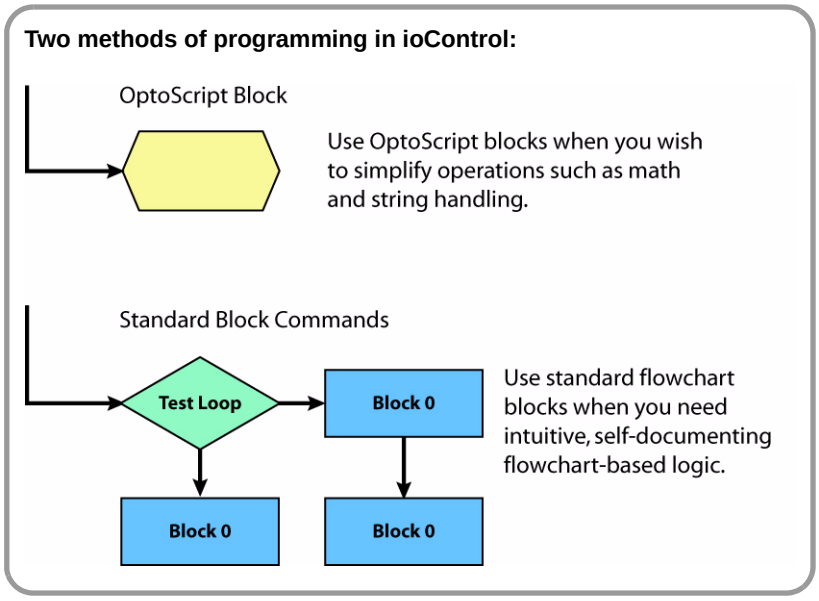
OptoScript

OptoScript blocks contain code written in OptoScript, a programming language that:

- is based on C, Pascal, and Basic
- offers the same commands used in other flowchart blocks
- provides online scripting assistance in choosing commands and strategy variables
- checks script syntax.

OptoScript is an optional programming language that can simplify certain types of operations in ioControl, giving you an alternative to using standard ioControl commands.

OptoScript does not add new functions, but offers an alternative method of programming. OptoScript code cannot be mixed with commands in Action or Condition blocks; rather, it is used in its own hexagonal block.



You'll want to use OptoScript for programming tasks that can be more difficult or time consuming using standard ioControl commands. For example, extensive math calculations and complex loops can take up a lot of space on a flowchart.

OptoScript is especially useful for the following:

- Math expressions
- String handling
- Complex loops
- Case statements
- Conditions
- Combining all of the above.

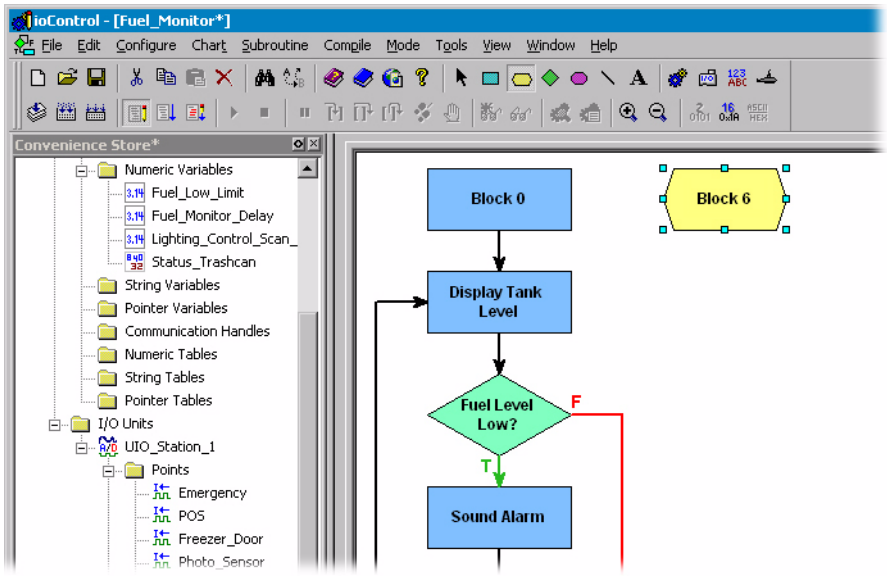
ACTIVITY

Creating an OptoScript Block

1. Make sure you are in Configure mode and the Fuel_Monitor chart is open.

2. Create a new block.

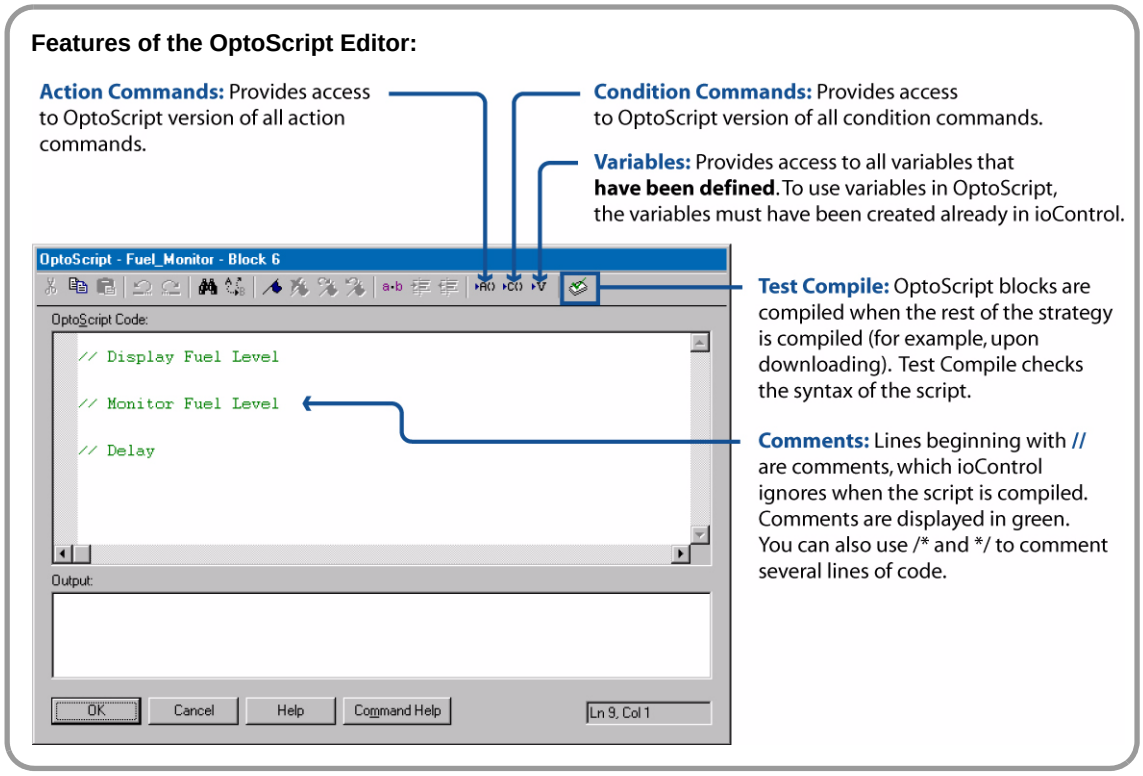
a. Click the *OptoScript Block* tool.
b. Click in the *Fuel_Monitor* chart to place the OptoScript block as shown:



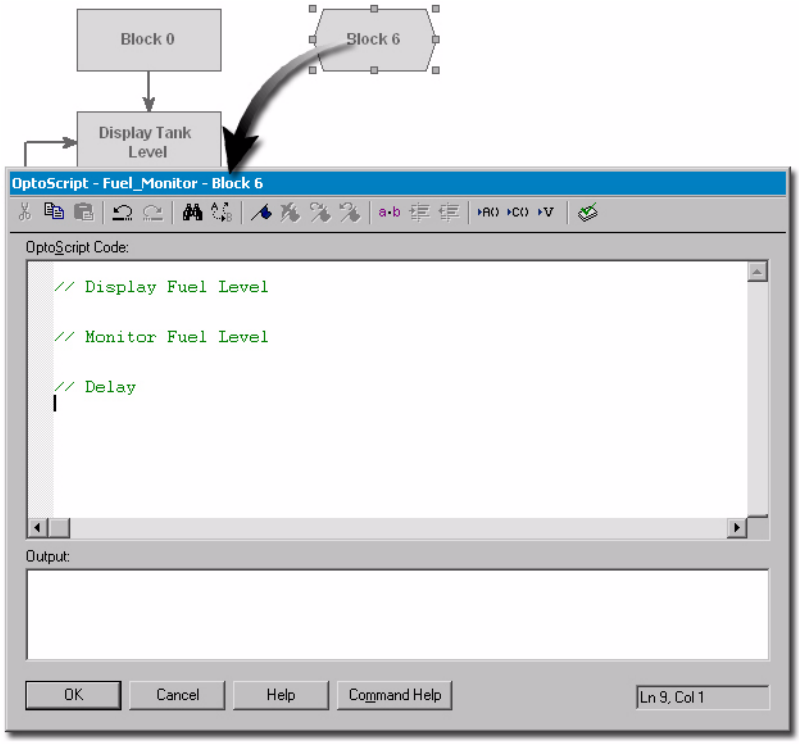
When we are finished with this block, it will replace the flowchart we created in the previous lesson. Typically, you would not program an entire flowchart in one OptoScript block. You would normally use OptoScript blocks in conjuntion with action and condition blocks. The purpose of this activity is to illustrate how to use OptoScript, and to show that, because it is more compact, it can replace several blocks.

3. Open the OptoScript block.

a. Click the *Select* tool.
b. Double-click the *OptoScript Block* to open the OptoScript editor.



c. Type the comments shown here. (Comments are very important in OptoScript blocks, enabling you and others to return to the code and quickly understand the programming logic.)

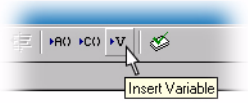


Using Variables in OptoScript

The entire strategy database, which includes I/O points, charts, and variables, is available to OptoScript. All are treated as variables within the OptoScript editor. Variables must be created in ioControl before using the OptoScript editor. You cannot create variables on the fly from within an OptoScript block. But, unlike other programming languages, you do not have to declare variables before you use them in an OptoScript block.

In this first line we want to move the value of the analog input point Fuel_Level to the analog output point Fuel_Display.

1. Insert the analog output variable, Fuel_Display.

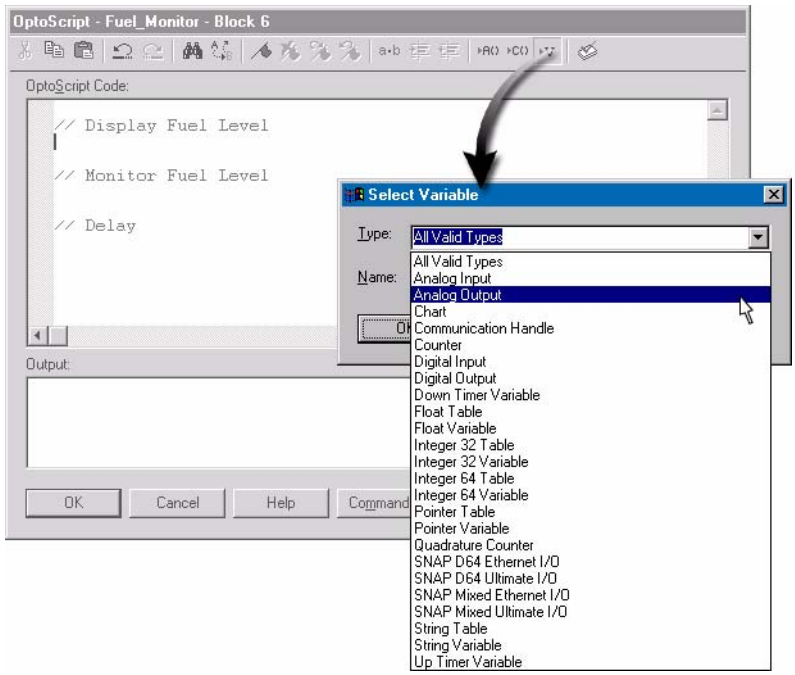


a. Place your typing cursor below the text `//Display Fuel Level`.

b. Click the *Insert Variable* button to open the Select Variable dialog box.

c. Select *Analog Output* in the *Type* field.

As Fuel_Display is the only Analog Output variable, it should be selected.

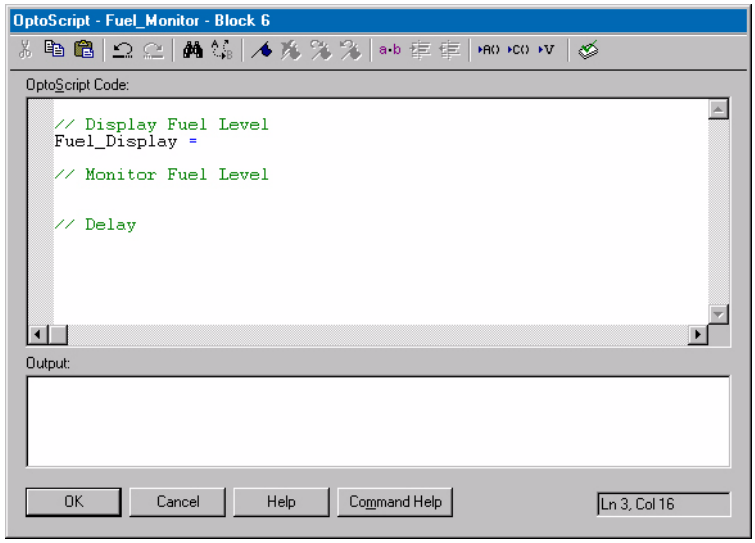


d. Click *OK*.

`Fuel_Display` will appear in the OptoScript Editor.

2. Insert the operator.

Type an equal sign (=).



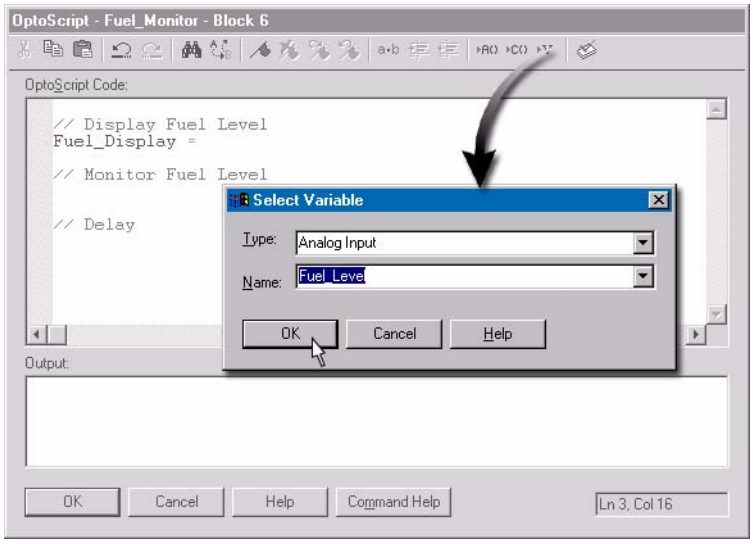
NOTE: Spacing is optional, but makes your code easier to read.

> **ioControl Commands and OptoScript**
>
> Some ioControl commands have a matching OptoScript command, which we'll use shortly. Some are accomplished without a command by using the built-in functionality. For example, instead of using an ADD command, you would just use a plus symbol (+). These are documented in the *ioControl User's Guide*, in the OptoScript Command Equivalents Appendix. If we looked in that appendix, we'd see that instead of MOVE, we use an equal sign (=).

3. Insert the analog input variable.

a. Make sure the typing cursor is after the equal sign.
b. Open the Select Variable dialog box.
c. In the *Type* field, choose *Analog Input*.

d. In the *Name* field, choose *Fuel_Level*.



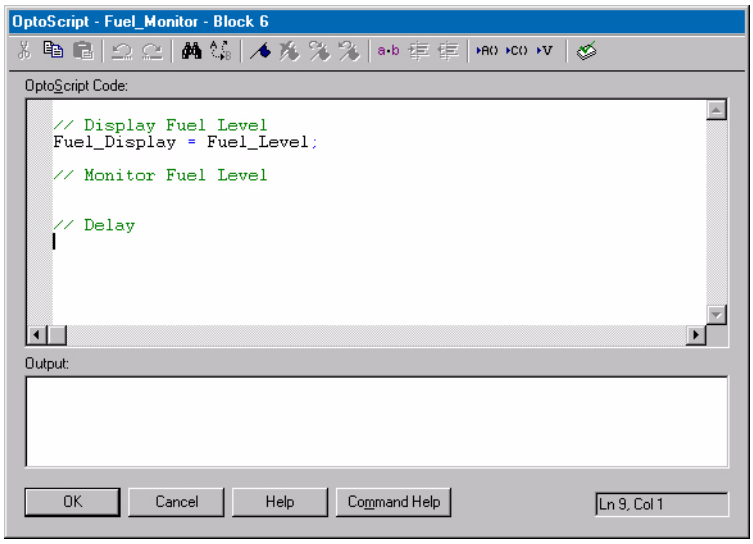
e. Click *OK*.
f. Type a semicolon at the end of the line.

Using Action Commands in OptoScript

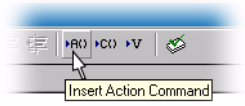
In the next series of steps you will learn how to select Action commands and to use the Command help. We will create a delay command to execute a five-second delay between loops that check the need to turn on the outside lights. (In a real application, we would use a longer delay, such as 15 minutes.)

1. Insert the action command.

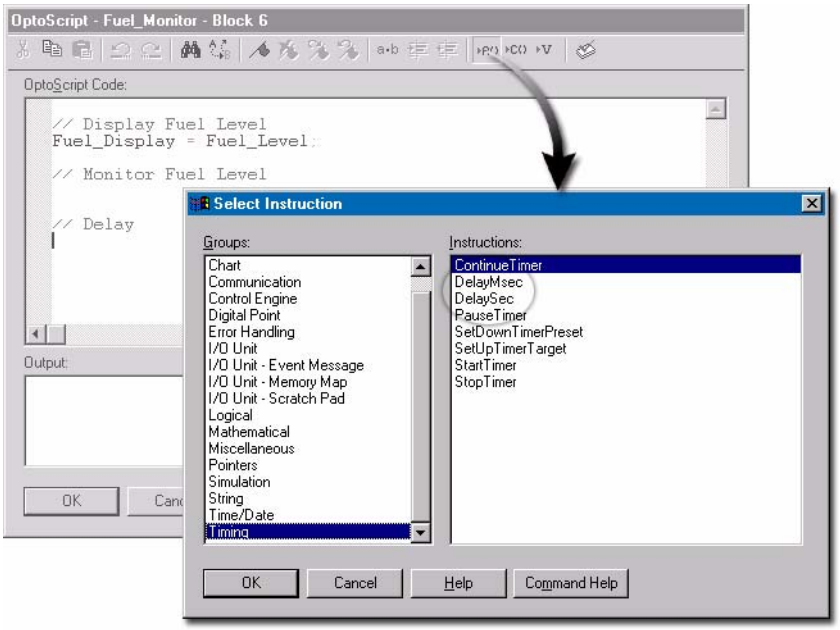
a. Move the typing cursor to the line below the //Delay comment. (Insert a linebreak if necessary.)



b. Click the *Insert Action Command* button.



c. In the *Groups* field, select *Timing*.
Notice that the there are two delay commands available.

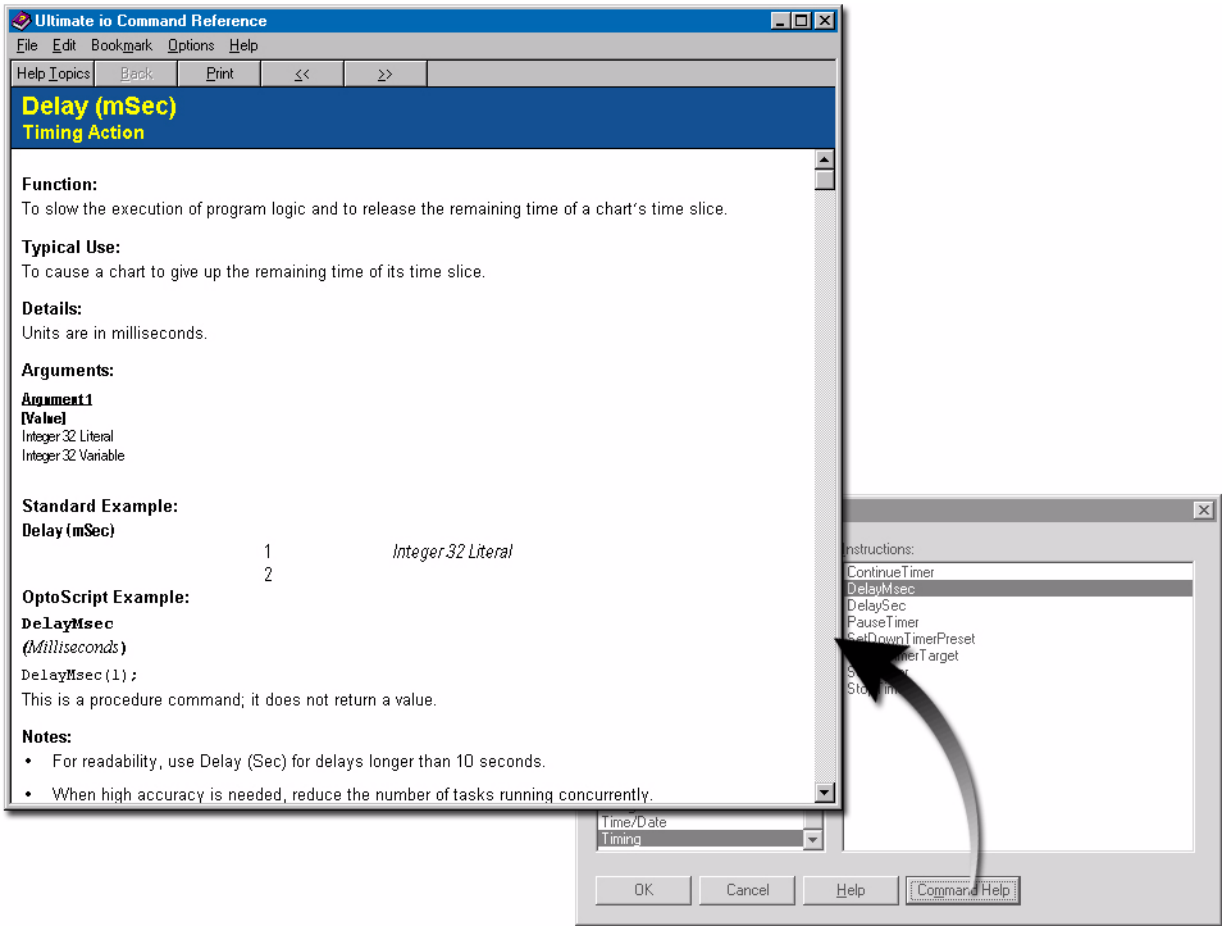


One of the delay commands is more suitable than the other for the delay we need. You can read the help for each command to determine which is better.

2. Use Command Help to determine the best command to use.

a. Select the *DelayMsec* command.

b. Click *Command Help*.



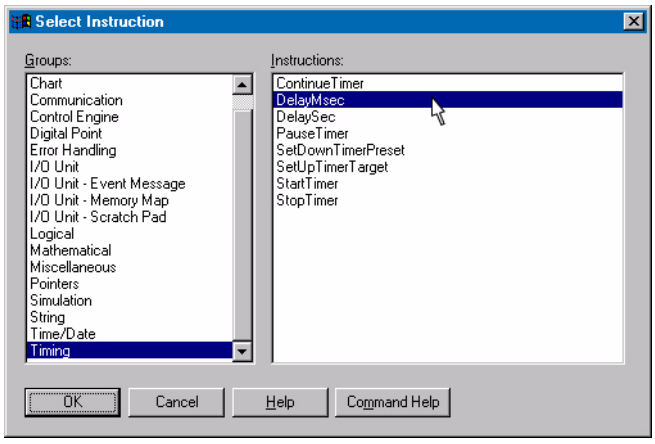
c. Scroll down the help window to the Notes section.

The help says "For readability, use Delay (Sec) for delays longer than 10 seconds." If we were creating a 15-minute delay, we would use Delay(Sec), but since we are using a 5-second delay, DelayMsec is the better command to use.

d. Close the Help window.

3. Insert the DelayMsec command.

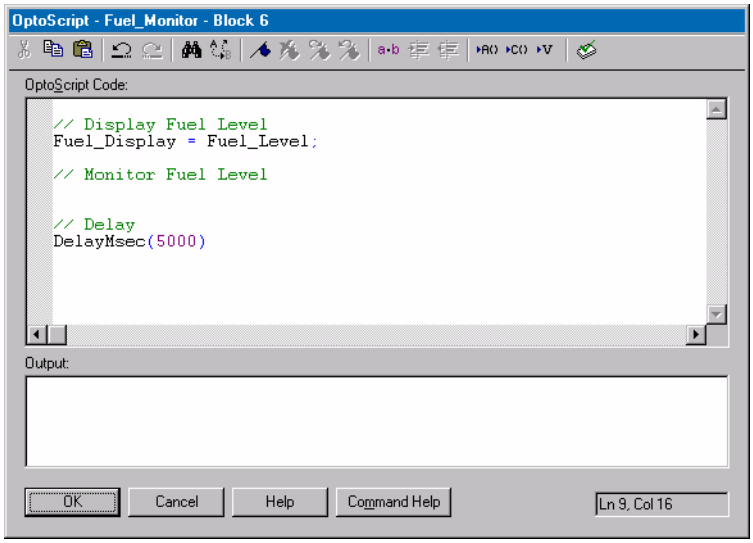
a. From the Select Instruction dialog box, select the *DelayMsec* command (if it is not already selected).



b. Click *OK*.

c. Add the parentheses and the milliseconds parameter. Your code should appear as shown here:

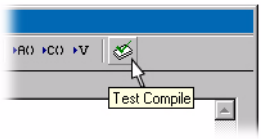
```
DelayMsec(5000)
```



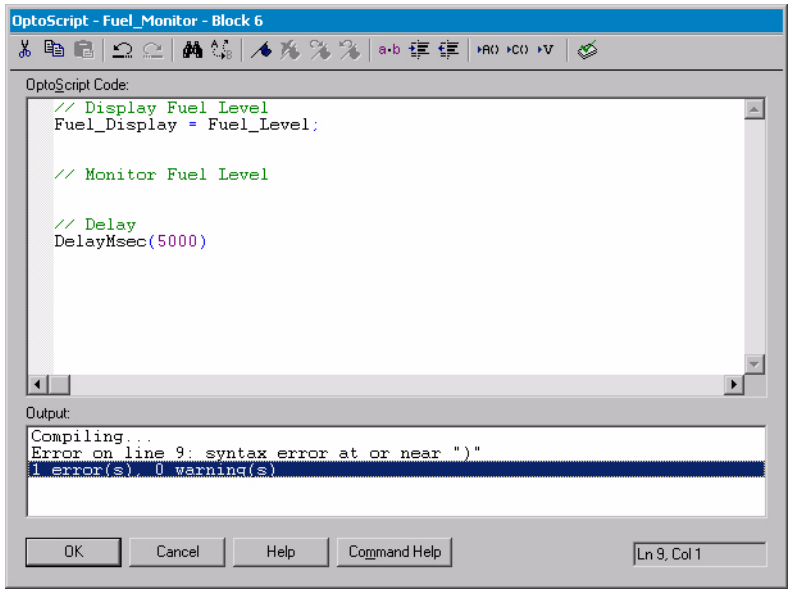
NOTE: We could have created an integer variable and used it in place of the 5000.

4. Test compile.

Click the *Test Compile* button.



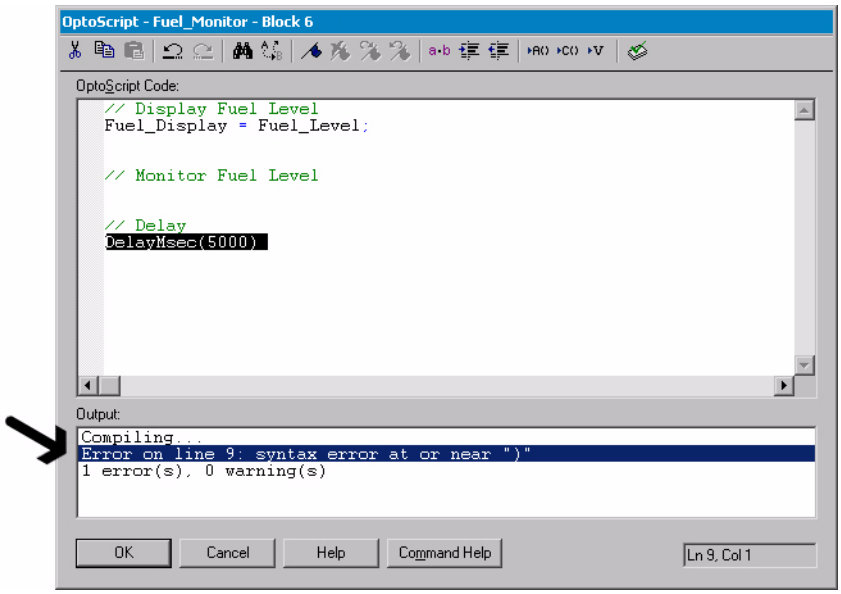
The DelayMsec command was incomplete and the test compiler reveals the error.



5. Use the Output window to locate an error.

The Output window will locate an error for you.

a. Double-click the error message in the Output window.

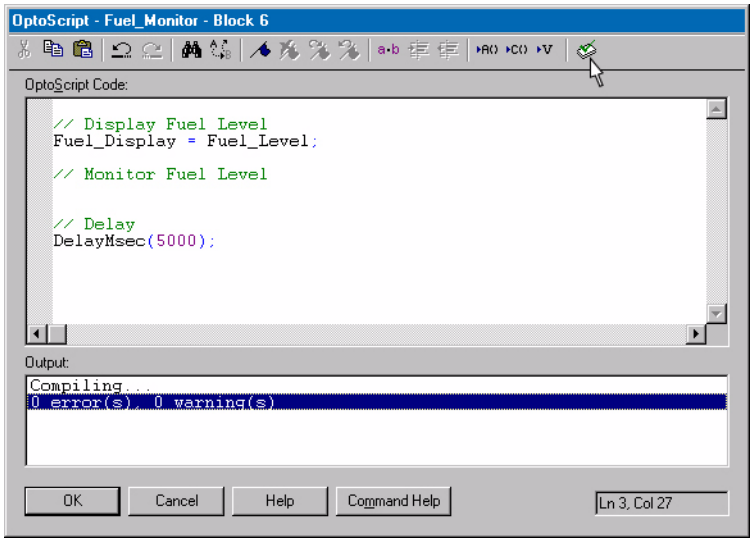


The source of the error is now highlighted. A semicolon is needed at the end of the DelayMsec command.

b. Click at the end of the DelayMsec command to remove the highlighting.

c. Type a semicolon.

d. Repeat the test compile.



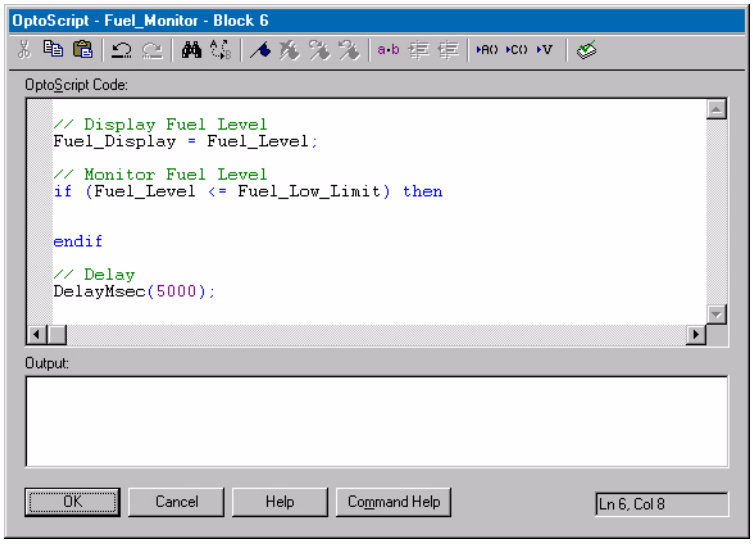
e. Keep the OptoScript editor open for the next series of steps.

Creating Conditional Statements

Like loops and case statements, conditional logic can be simpler in OptoScript than in flowcharts. If/then, if/then/else, and if/then/elseif statements can all be mixed and nested within OptoScript as needed. (Note: The syntax is documented in the *ioControl User's Guide*, Opto form #1300. Sample code is also included in the ioControl example strategies.)

1. Create an if/then statement.

Type the partial if/then statement shown below:



Summary of Steps

1. Create the if/then structure:

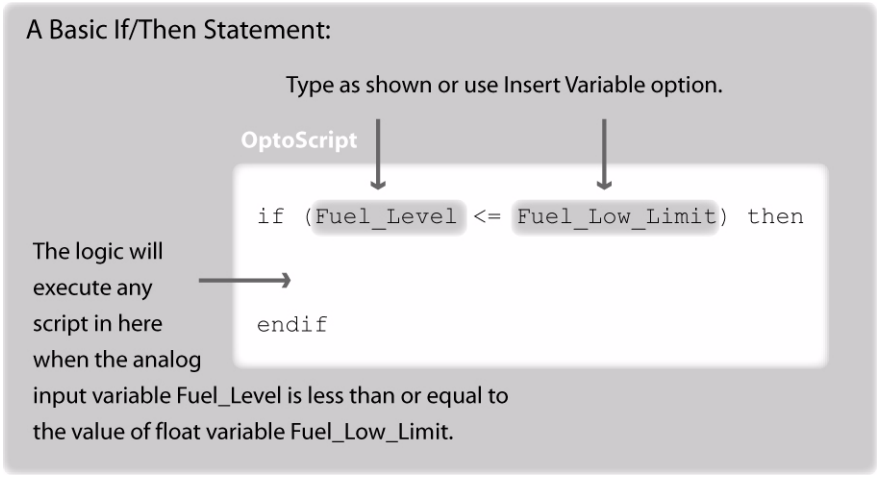
```
if ( ) then


endif
```

2. Type or use the Select Variable dialog box to insert the following variables:

- Fuel_Level
- Fuel_Low_Limit

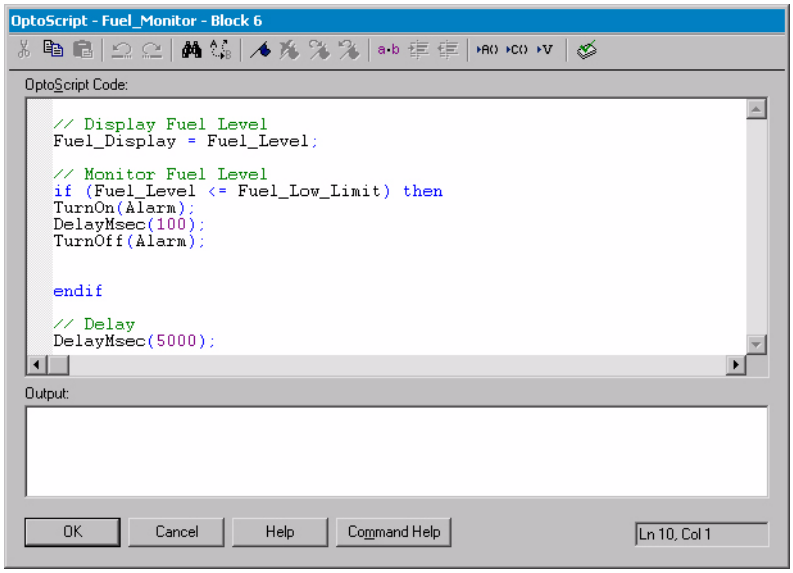
3. Type `<=` for less than or equal to.

NOTE: Using Select Variable helps prevent spelling errors.



2. Insert an action command.

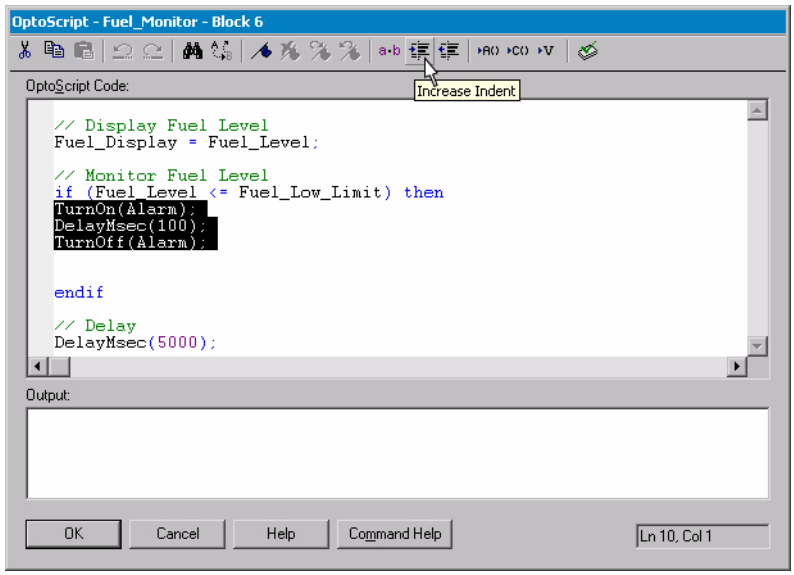
Complete the if/then statement as shown:



3. Increase indent.

The OptoScript editor can adjust the indenting of text to make the code easier to understand.

a. Highlight the lines shown below.



b. Click the *Increase Indent* button.

c. Do a test compile ( ).

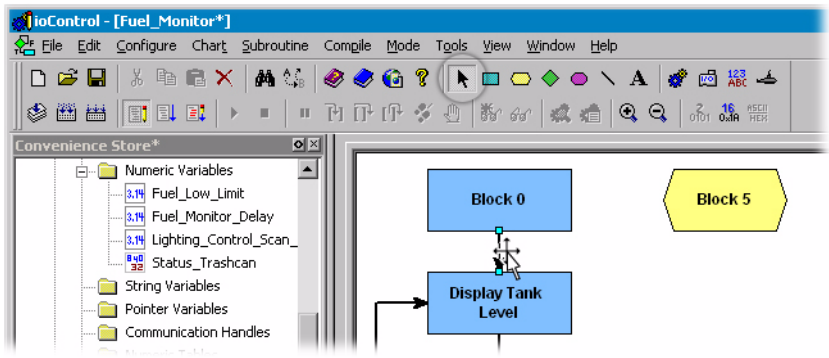
d. Click *OK* to close the OptoScript editor.

Modifying the Flowchart

Your OptoScript block is ready to replace your chart.

1. Connect the OptoScript block.

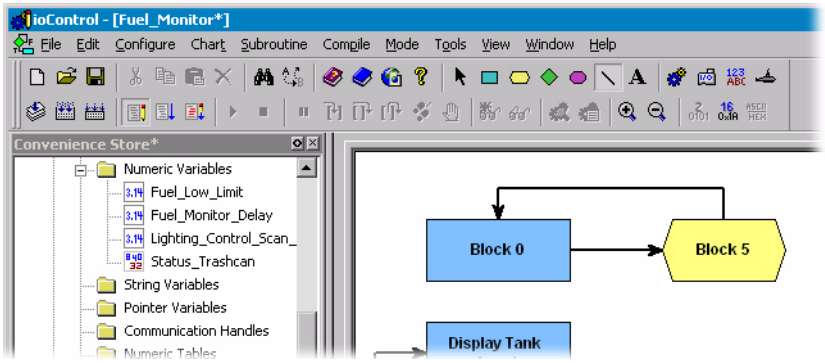
a. Select the connection line between Block 0 and Display Tank Level.



b. Press the *Delete* key.

c. Select the *Connect* tool.
d. Click *Block 0* (see diagram below).
e. Click the OptoScript block.
f. Click the OptoScript block again to start the second connection.
g. Click above the OptoScript block (this draws the connection line out of the top of the script block).
h. Click above *Block 0*.
i. Click *Block 0* (this completes the connection).

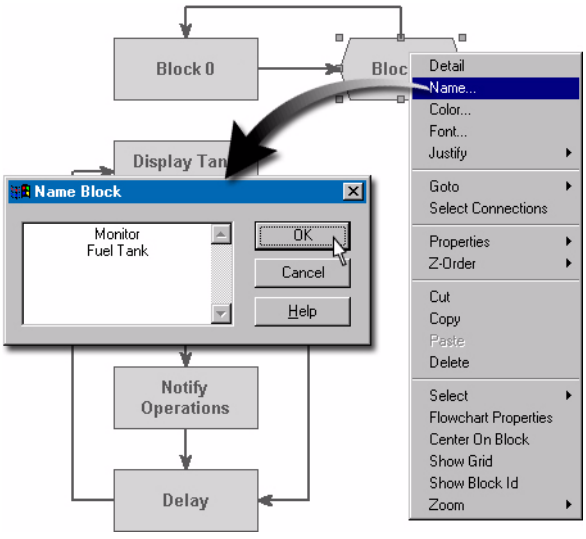
Review: Connect Tool



It is important to loop the connection lines. This enables the script to run continuously. Without looping back to the starting block, the logic would run the script only once, because the script block itself contains no looping instructions.

2. Rename your new block.

a. Click the Select tool.
b. Right-click your OptoScript block.
c. Click *Name*.
d. Type `Monitor Fuel Tank`.

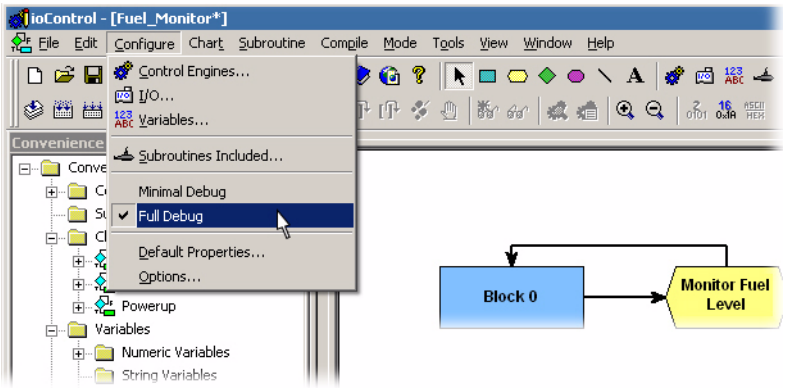


Debugging OptoScript

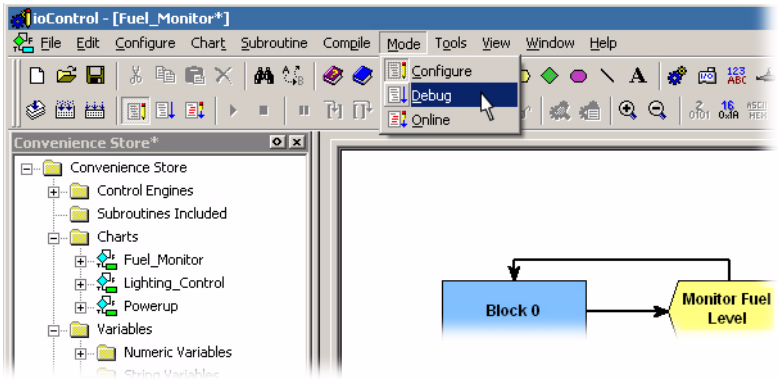
Debugging your strategy has new challenges now that you have a script block with several lines of code. We've been running Debug mode with minimal debug features. Now that we have OptoScript to test, we'll need to use the full set of debug features.

1. Change debug settings.

Choose menu *Configure* ➞ *Full Debug*.



2. Download the strategy.

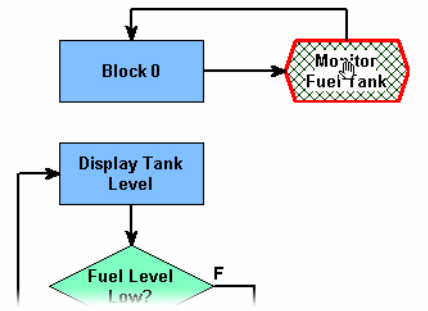


3. Run the strategy.

The chart should work just as before.

4. Substep through the Fuel Monitor chart.

a. With your chart still running, place a breakpoint on the Monitor Fuel Tank block.



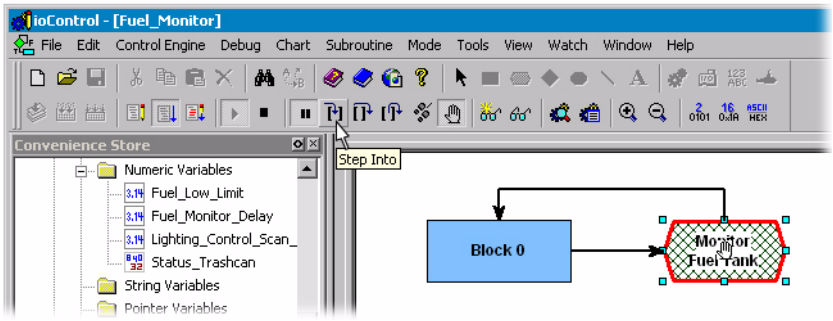
Review: Setting a Breakpoint

1. Click the *Breakpoint* tool while your chart is running.

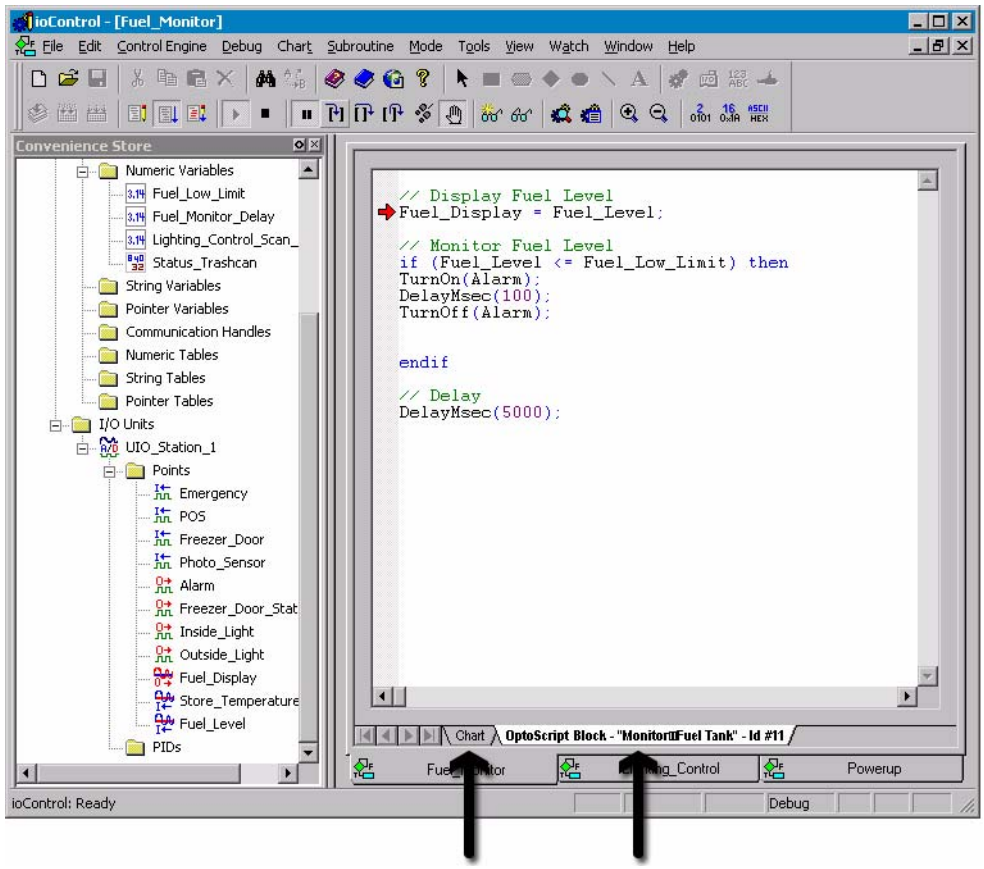
2. Click the *Monitor Fuel Tank* block.

b. Wait for the strategy to stop at the script block.

c. Click the *Step Into* button.



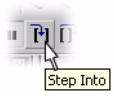
This button opens the OptoScript block window.



Tabs at the bottom of the window allow you to switch between the OptoScript block and the rest of the chart. The OptoScript block window remains open as along as the program execution is within this block.

Within the OptoScript block, a red arrow identifies where the program execution is. Clicking the Step Into button advances the program through individual commands. The Step Over button will advance through looped statements. The Step Out button will advance to the next block.

d. Click the *Step Into* button to see the execution continue line-by-line.

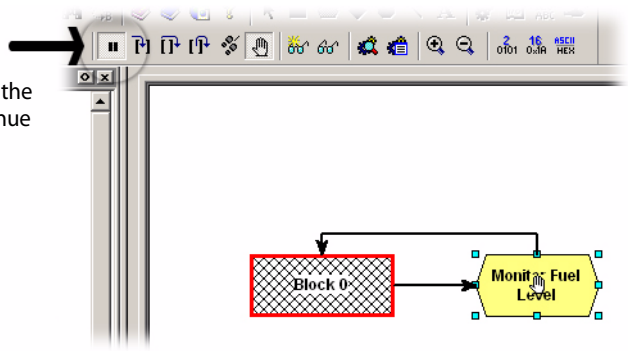


e. Click *Step Out* to exit this block.

You will notice a pause before the OptoScript window is closed. This is because the remaining lines in the OptoScript block include a 5-second

pause. When the OptoScript block is closed, the programming execution stops at the next block, which is Block 0.

The program execution is still paused, waiting for Step commands to continue. Alternatively, you can remove the paused state, which will continue to the next breakpoint.



5. Remove Auto Stepping.

a. Remove the breakpoint on your OptoScript block. (With the Breakpoint tool selected, click the OptoScript block.)
b. Click the pause button.

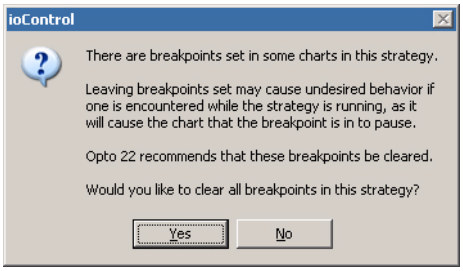
Returning to Configure Mode

1. Leave your strategy running.

You will need your strategy running for the next lesson on ioDisplay.

2. Return to Configure mode.

a. Choose menu command *Mode* ➞ *Configure.*
b. Click Yes to take the chart out of Step mode.



3. Save your strategy.

4. Close ioControl.

This completes the ioControl portion of the tutorial. While the instruction so far is not comprehensive, it does introduce you to many important concepts and features. To learn more, consult the *ioControl User's Guide* or the Opto 22 Web site. Of course, the best way to continue learning is to begin programming your own applications.

Operator Interfaces

ioDisplay Skills

Creating a Project

- Configuring a control engine
- Connecting to an ioControl strategy

Simple Graphics and Text

- Importing bitmaps
- Importing Symbol Factory graphics
- Drawing shapes
- Creating text
- Duplicating graphics

Formatting

- Aligning graphics and text
- Changing font styles

Dynamic Graphics and Text

- Using analog values to resize and recolor graphics
- Displaying analog values as text
- Using digital values to control the appearance of graphics
- Displaying digital states as text
- Writing to a digital point

Scenario

Your ioControl strategy is complete and running on the SNAP Ultimate I/O system in your convenience store. The following capabilities are available:

- The inside lighting is turned on when the strategy starts running.
- The photo sensor triggers the outside lighting.
- The freezer door switch controls the freezer door status light.
- The emergency button triggers the alarm.
- The tank meter displays the value of the tank level potentiometer using a scale of 0–10,000 gallons.

The operation of your SNAP Ultimate I/O unit is independent of your workstation. You used your workstation to monitor the programming and to debug the code. Though you can follow the control logic using Debug mode, ioControl is not a suitable interface for an operator who monitors a process. It's much more useful to create graphical and text displays in ioDisplay that represent your system. In this lesson, you use your workstation as part of a process monitoring system that you build.

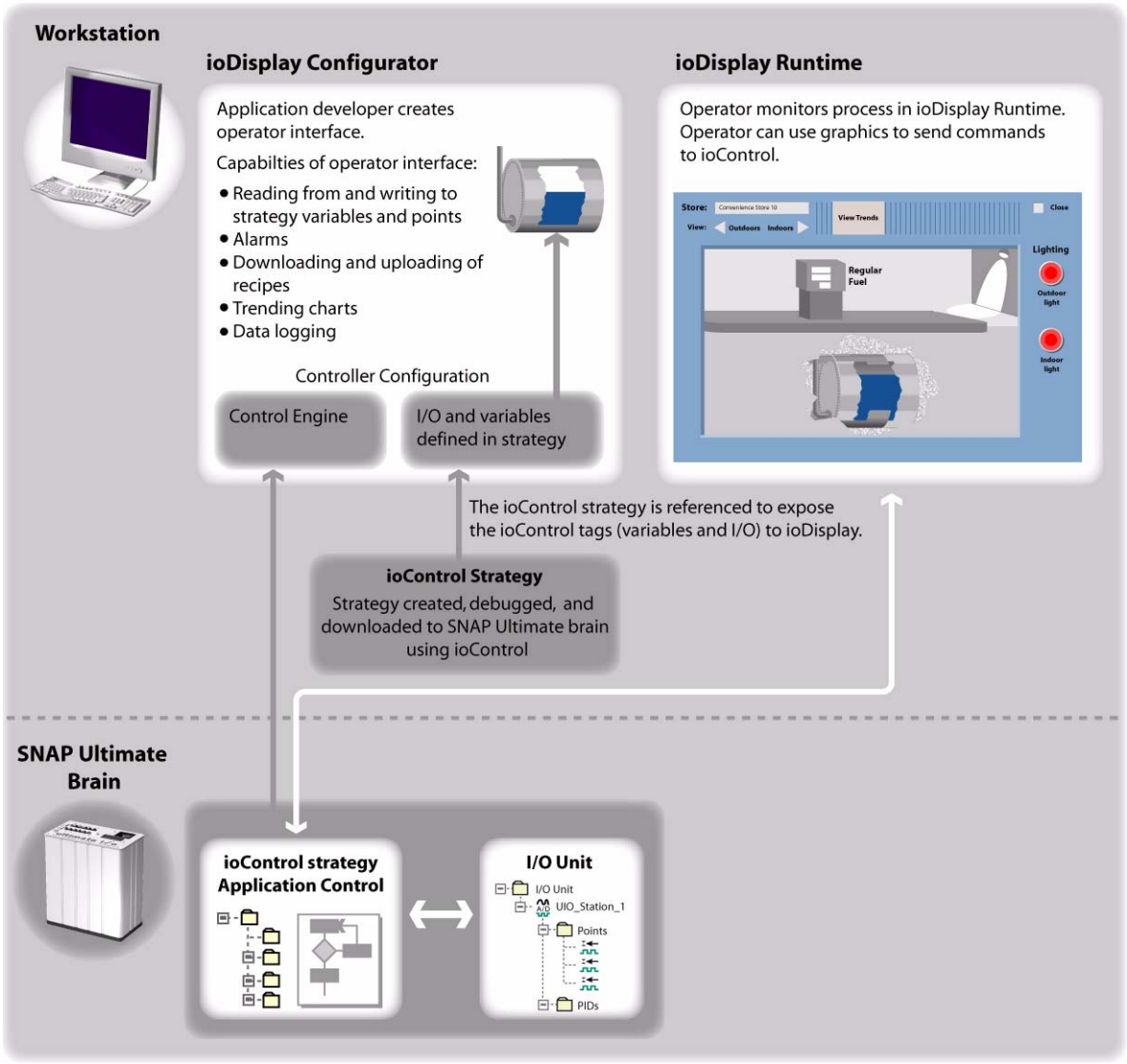
ioDisplay is a process visualization program that reads from and writes to the tags in one or more ioControl strategies. The result is a graphical user interface that you design to meet your needs, whether you are trying to visualize a complex process or summarize the key elements of one in an operator interface. This type of tool is often referred to as a monitoring application, or a human–machine interface (HMI).

In this lesson, you create a graphical user interface to monitor the following from your convenience store:

- Level of the fuel tank
- State of indoor and outdoor lighting.

CONCEPTS

ioDisplay within SNAP Your Ultimate I/O System



ioDisplay Capabilities

ioDisplay is a software application specifically designed for use with ioControl. ioDisplay allows easy interaction between a process operator and the ioControl strategy running on a SNAP Ultimate controller/brain. ioDisplay can interface with a single control engine or with multiple control engines, each running a different ioControl strategy.

ioDisplay connects to a tag database, which contains the readable and writable elements in an ioControl strategy. By connecting tags to graphical objects that you draw, you can create on-screen animations that represent the process running on your SNAP Ultimate I/O unit and provide operator controls. You can configure a graphic to reflect the state or value of a tag, write to a tag, or both. For example, you can use the

value of an analog point to change the color, size, position, or visibility of a graphic. You can also use graphics that the operator can select to write changes to tags, for example, clicking on a graphic of a light switch to turn on a light.

Connecting graphics to tags is a simple process due to ioDisplay's tight integration with ioControl. This integration eliminates the need to retype data into a tag dictionary, a requirement with many other HMI programs. Other ioDisplay capabilities include real-time trending, historic data logging, application launching, file uploading and downloading, and multimedia support.

Control vs. Monitoring

Control and monitoring (or interfacing) are two very distinct aspects of your system. Opto 22 draws a line that separates the control of your system from the user interface that interacts with your system.

Control

ioControl is a process control language. The strategy that you develop is used to control your system.

Monitoring

ioDisplay is a software application used to create an operator interface for an ioControl strategy. ioDisplay does not control your system. It is an interface tool. The project that you develop using ioDisplay monitors the ioControl strategy as it is running. ioDisplay can be used to change the values of tags in the ioControl strategy, but it does not contain programming or logic-solving capabilities to actually control the process.

Configurator vs. Runtime

ioDisplay has two separate programs: Configurator and Runtime.

Configurator

The Configurator defines and configures the ioDisplay windows, graphics, dynamic attributes, logs, alarms, and sounds that make up your project. The objects are connected to variables, I/O points, PID loops, or other tags defined in your ioControl strategy. Every ioControl strategy stores information on global data items for that strategy in a file with the extension *.IDB. ioDisplay can link to any ioControl strategy through its *.IDB file.

Runtime

ioDisplay Runtime communicates with the processor running the ioControl strategy. Runtime animates the operator interface by updating the graphic attributes (color, size, position, and visibility) in response to changes in the ioControl tags.

Objects

An ioDisplay project consists of many objects. Objects include drawing windows, graphics, alarm triggers, and trends. Objects can be either static or dynamic.

Static Objects

Static objects are not attached to ioControl tags and have no dynamic attributes. Their appearance does not change while ioDisplay Runtime is running. Examples of static objects are labels and backgrounds.

Dynamic Objects

Dynamic objects have attributes that can change the appearance of objects in ioDisplay. Attributes can also change the value of variables or states of I/O points in an ioControl strategy. In other words, dynamic attributes can be used to read tags, write to tags, or both.

Tags

A tag represents data items (such as points and variables) that have been configured in an ioControl strategy.

To animate your operator interface, ioDisplay graphics are connected to tags and assigned graphic dynamic attributes. Dynamic attributes are used to change the appearance of the graphic in response to a tag's change in value (control engine-driven attribute), or are used to change the value of a tag (operator-driven attribute).

Tags are also used as triggers to initiate system events such as sounds, historic logging, and window configurations.

ACTIVITY

Preparing for the Lesson

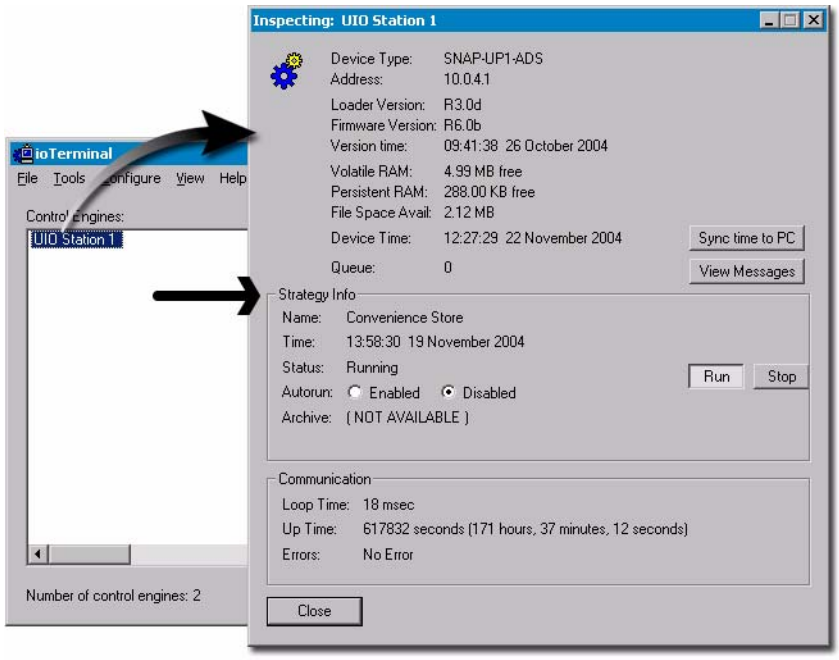
Make sure your ioControl strategy is loaded and running on the SNAP Ultimate controller/brain.

You can use the ioProject utility ioTerminal to check whether your strategy is running:

1. Start ioTerminal.

From the Start menu, select
Programs ⟶ *Opto22* ⟶ *ioProject Software* ⟶ *ioUtilities* ⟶ *ioTerminal*.

2. Double-click the name of the control engine to open the Inspecting window.



The name and operational state of the strategy is shown in the Strategy Info group. A running strategy is indicated by Running in the Status field and the Run button is selected. You can use the Run button to start the strategy if it is not running.

Creating an ioDisplay Project

1. Launch ioDisplay Configurator.

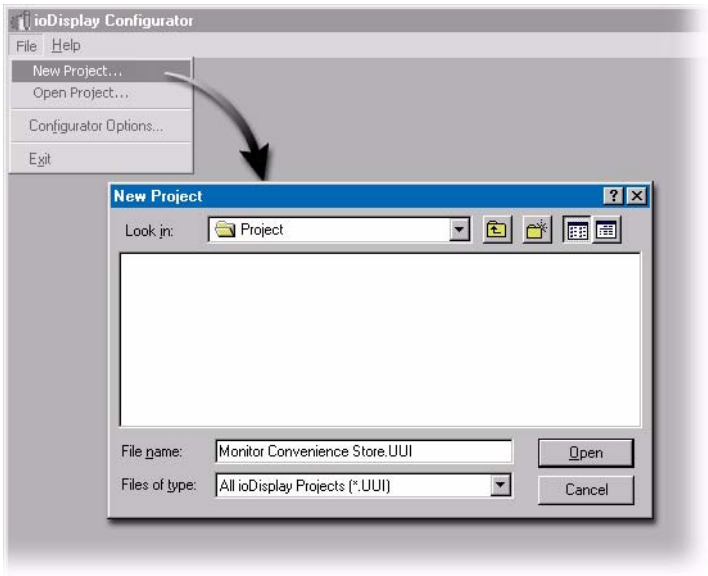
From the Start menu, select

Programs ⟶ *Opto22* ⟶ *ioProject Software* ⟶ *ioDisplay* ⟶ *Configurator*.

2. Create a new ioDisplay project.

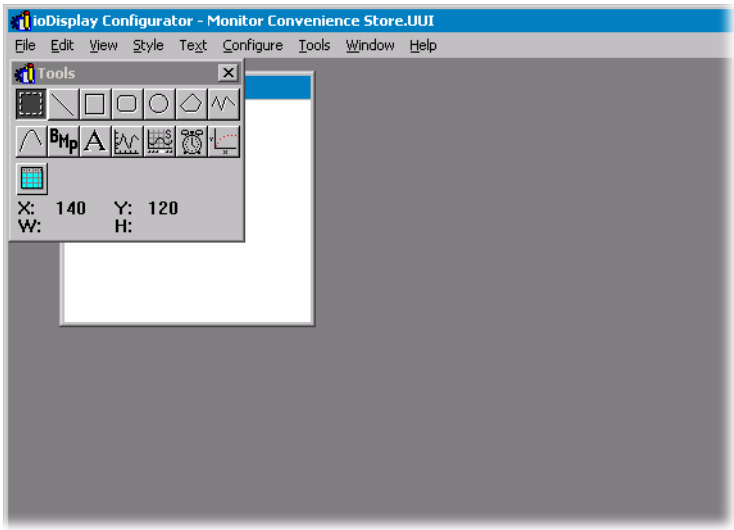
a. Select *File* ⟶ *New Project* to open the New Project window.

b. Navigate to the **C:\Program Files\Opto22\UIO Learning Center\Project** folder.



c. Type `Monitor Convenience Store` in the *File name* field. This will be the name of your ioDisplay project.

d. Click *Open*.



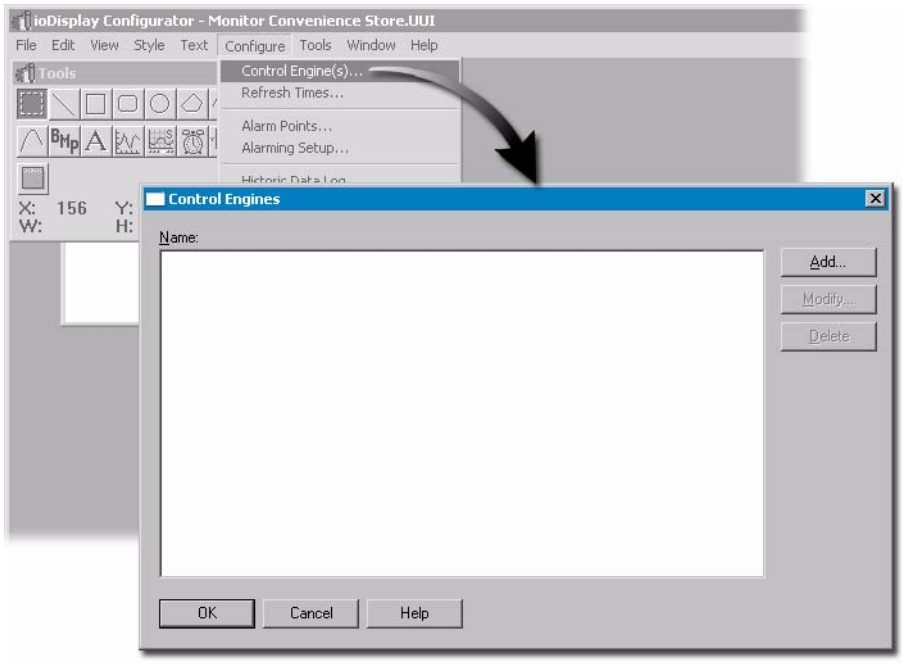
Linking the Control Engine

The building blocks of ioDisplay projects are the I/O point and variable tags used by an ioControl strategy. You gain access to these tags by linking the project to a control engine.

Linking to a control engine is a two-step process: 1) you specify the ioControl strategy running on the control engine; then 2) you select or create the control engine configuration on the PC.

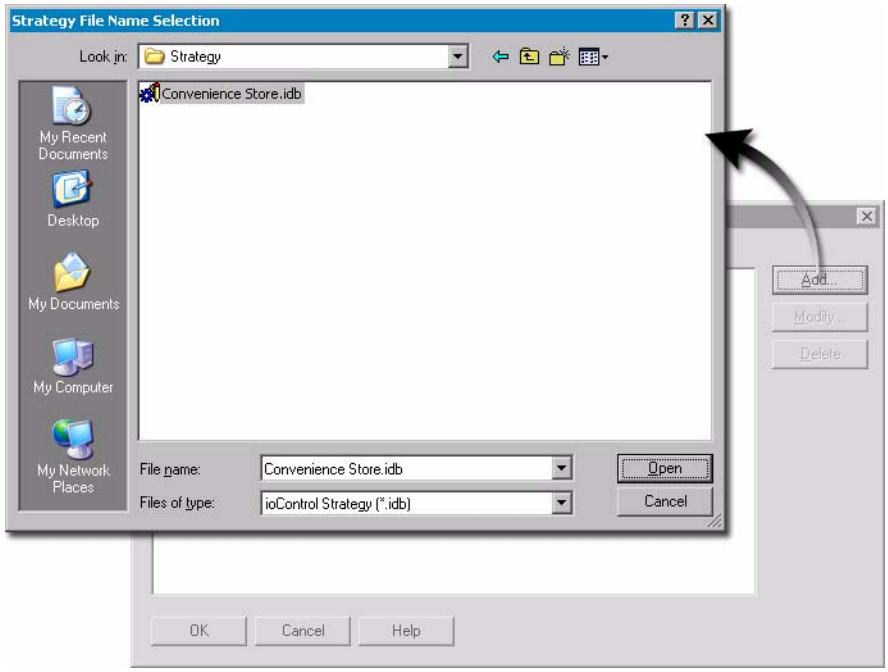
1. Open the Control Engines dialog box.

Select *Configure* ➞ *Control Engine(s).*

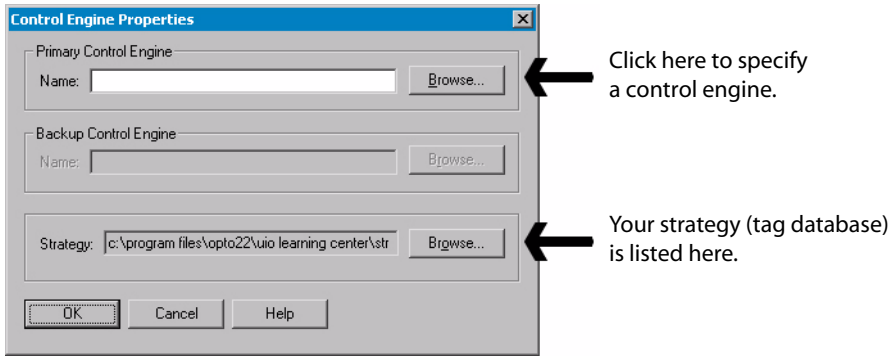


2. Open the ioControl tag database.

a. In the Control Engines dialog box, click *Add.*
b. Move to the **C:\Program Files\Opto22\UIO Learning Center\Strategy** folder.
c. Select and open *Convenience Store.idb.*

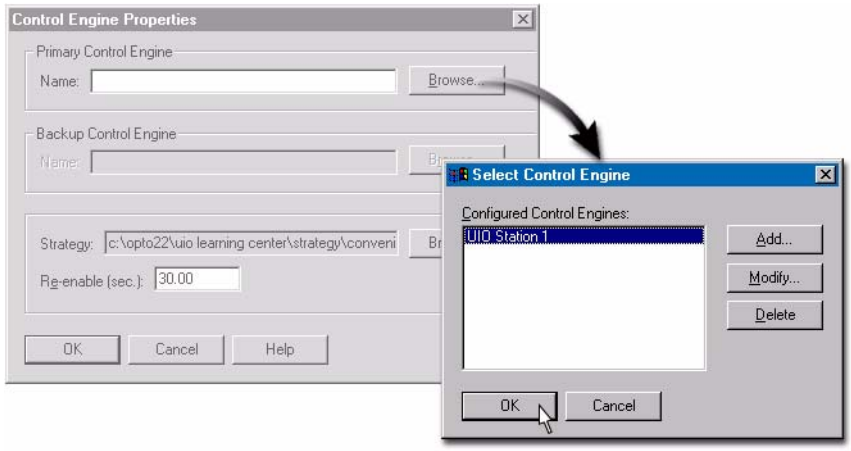


After providing the tag database, you are prompted to specify the control engine.



The *.idb file is a database of all tags used in the specified ioControl strategy. ioDisplay links directly to the file. There is no need to re-enter tag names.

d. Click the *Browse* button to select a Primary Control Engine.



e. Select the *UIO Station 1* control engine.
f. Click *OK* to close the Select Control Engine dialog box.

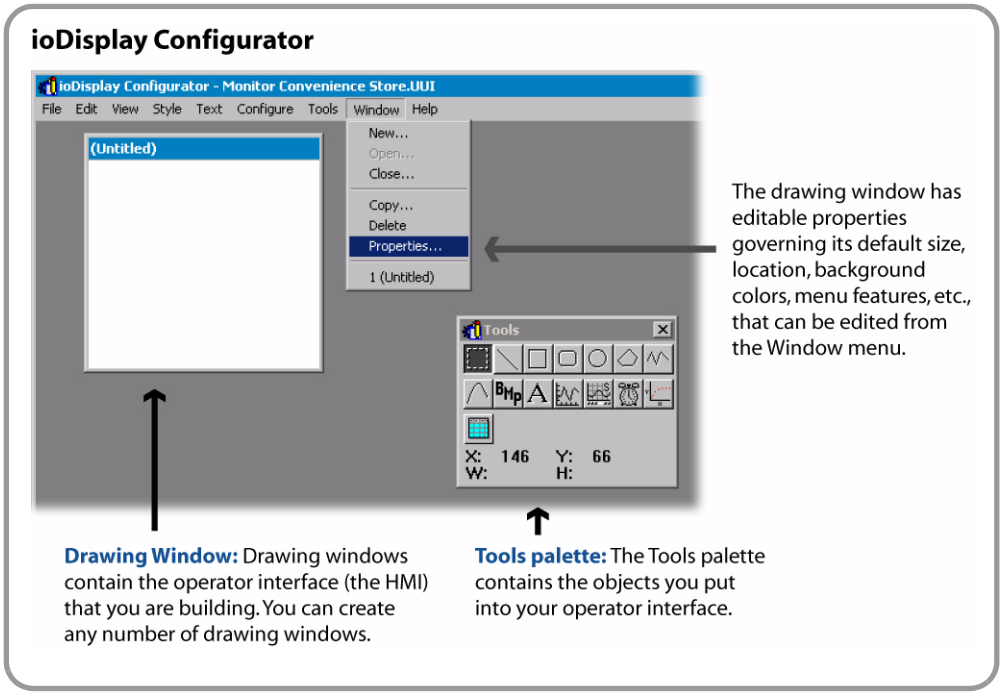
3. Close the Configure Control Engine dialog boxes.

a. Click *OK* to close the Control Engine Properties dialog box.
b. Click *OK* to close the Control Engines dialog box.

Changing the Properties of the Drawing Window

Your new project contains a drawing window, which will become part of the operator interface you are building. The Tools palette provides access to the various ioDisplay objects you can create.

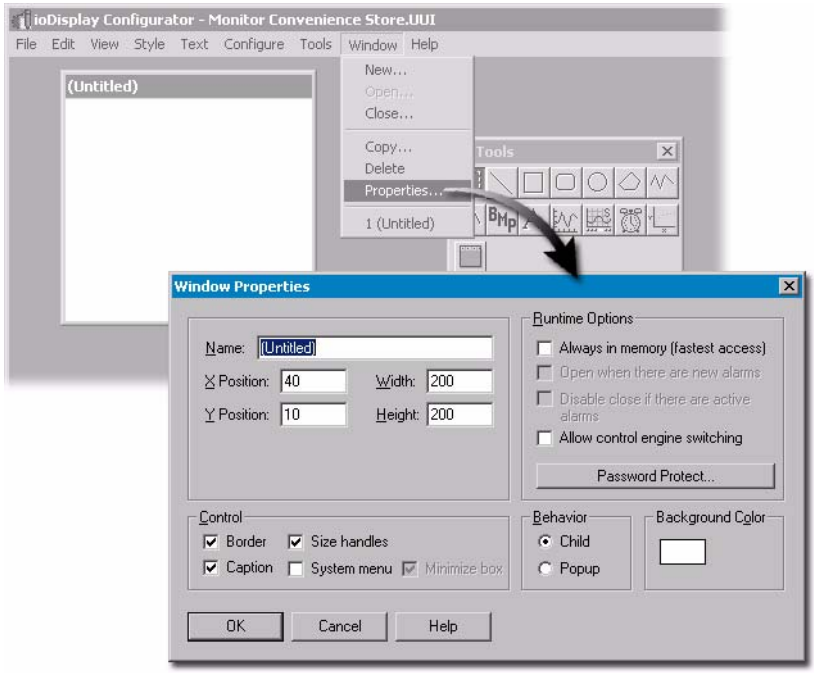
If the Tools palette is covering your drawing window, drag the Tools palette to another part of the application window.



1. Open the Window Properties dialog box.

The properties of the drawing window describe the size, color, position, and behavior of the window.

Select *Window* ⟶ *Properties*.



2. Name the window.

Type `Main` in the *Name* field.

Every window that you create in the project has its own unique name. You will use this window as the main window in your project. In this window, you can create buttons to open and close other windows.

3. Edit window properties.

a. Type `0` in both the X Position and Y Position fields.

The position 0,0 indicates that the top-left corner of the window will be placed at the top-left corner of your computer screen.

b. Type `800` in the Width field.

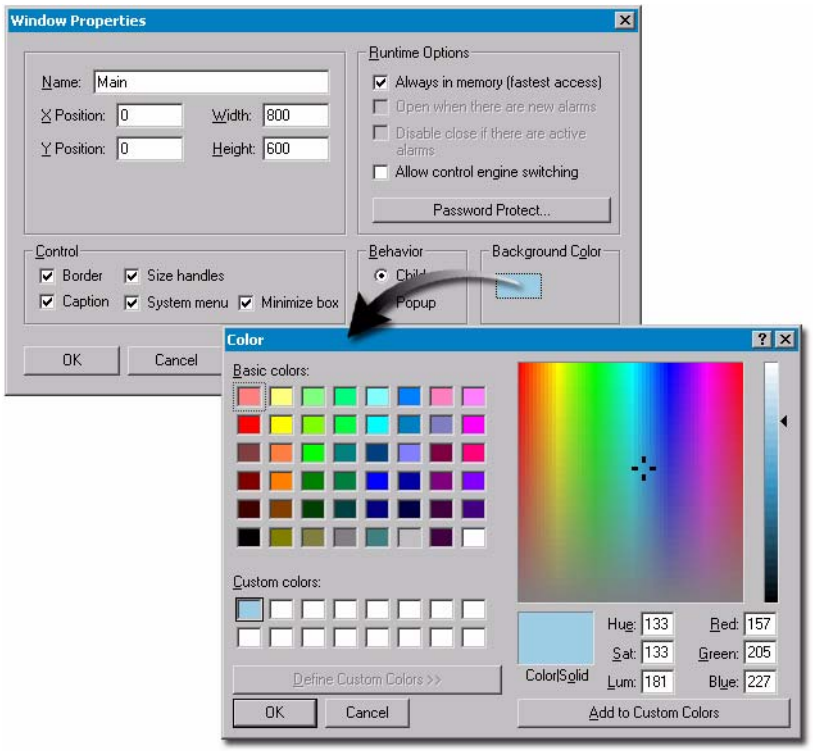
c. Type `600` in the Height field.

Specifying 800 x 600 creates a full-screen window when the resolution of the monitor is 800 by 600. The window can be enlarged later if the specified dimensions do not produce the desired size on your workstation. (However, it is difficult to reduce your project if it gets used on a system of lower resolution.)

d. Click the *System menu* option box so that it is checked.

This option provides Minimize and Maximize buttons, along with a System Menu button on the left side of the window.

e. Click the *Always in memory (fastest access)* option box in the Runtime Options field.



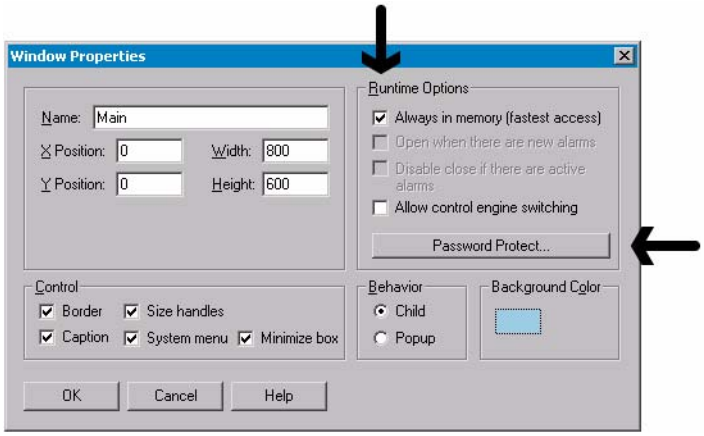
f. Click the *Background Color* box.

g. Select a light color by clicking on it.

h. Click *OK* to close the Color dialog box.

i. Click *OK* to close the Window Properties dialog box.

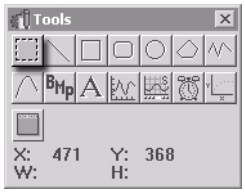
Always in memory specifies that this window's information should be loaded and always saved in the ioDisplay computer's memory during Runtime. Normally, only opened or iconified windows are saved in memory. Use this option for a window you know will be opened and closed often, as it will save time when the window is opened. (There are trade-offs when using this option: the more windows you have with this option selected, the more computer memory is used and the more time your initial ioDisplay setup will take. Having fewer windows with this option uses less computer memory, speeds up the initial ioDisplay setup, but costs you time when a window is opened.)



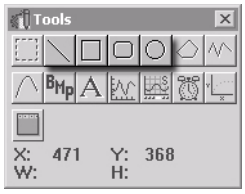
The **Password Protect button** lets you assign a password to a window. When a password is assigned to a drawing window, a closed window cannot be opened until the operator supplies the password. Open windows that are iconified or hidden are not affected. You cannot assign a password to a window that both contains an alarm and has the Runtime option "Open when there are new alarms" selected.

Examining the Tools Palette

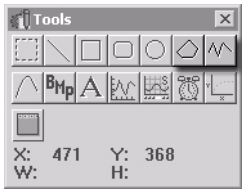
Now, you will begin drawing and selecting objects for your user interface. Take a few moments to examine the *Tools* window and become familiar with the function of each tool.



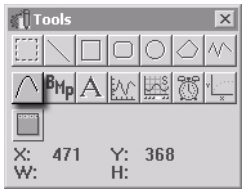
The **Select** tool is used to select one or more objects in the drawing window. Objects must be selected before you can change their properties, attach tags, or delete or move them.



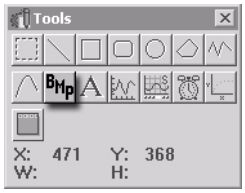
The **Line, Rectangle, Round Rectangle,** and **Ellipse** are used to create the shape indicated by the tool name. To draw the object, position the cursor in the drawing window; press and hold the left mouse button as you drag the mouse and form the desired shape. To draw a straight horizontal or vertical line, a perfect square (or square with rounded corners), or an exact circle, hold the `Ctrl` key down while drawing the object.



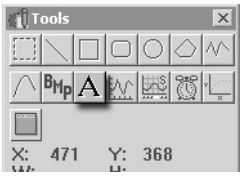
The **Free-Style Polygon** and **Free-Style Line** tools are used to draw complex shapes and lines. Apply a click-and-drag motion to draw and complete the object by double-clicking the last point you draw. The easiest way to draw a precise shape is to use one of these tools to establish the number of points, and then use the Select tool to right-click the object and choose *Edit Points*.



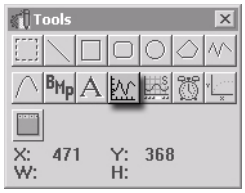
The **Free-Style Curve** tool draws curves that you can reshape. Draw a curve by clicking four places approximating your curve. The first click establishes the first end point. The second click places an attractor that governs the curve from the first end point. The third click places an attractor for the fourth endpoint. To edit the curve, use the Select tool to right-click your curve and choose *Edit Points*.



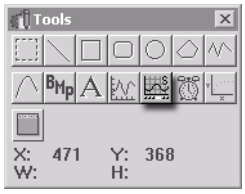
The **Bitmap** tool is used to paste selected bitmaps. To use this tool, you must first load a bitmap using *File* ➞ *Choose Bitmap*.



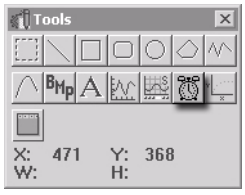
The **Text** tool is used to create text in a drawing window. Text can be a static object (a label) or a dynamic object (display a value). To use this tool, you 1) select the tool, 2) click in a drawing window, 3) type the text, 4) and click again in the drawing window. To edit text, use the Select tool to select and right-click the text.



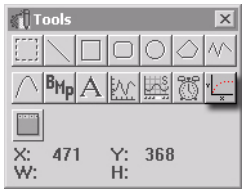
The **Trend** tool is used to create real-time trends. As with all dynamic objects, the attributes of the trend are determined after the trend is drawn in the window.



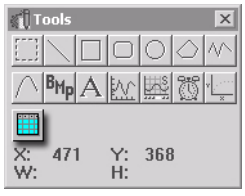
The **SuperTrend** tool is used to create trends that can be real-time, historical, or both. The SuperTrend has all the features of a regular trend, plus many more. SuperTrends can be disabled or hidden.



The **Alarm** tool is used to create alarm graphics. As with all dynamic objects, the attributes of the alarm graphic are determined after the graphic is drawn in the window. These graphics can be used to display notifications of alarms.



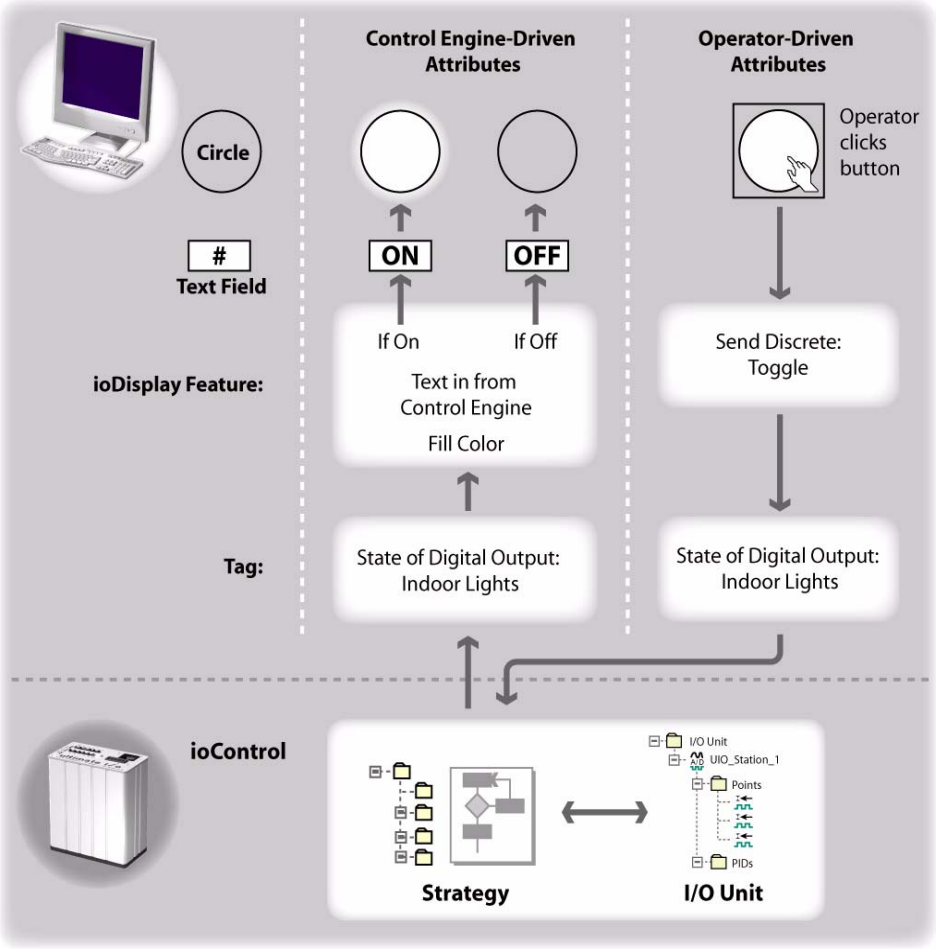
The **XY Trend** tool is used to create an XY plot from two numerical tables of data.



The **Numerical Table** tool creates a table that displays the contents of ioControl float tables, integer 32 tables, and integer 64 tables.

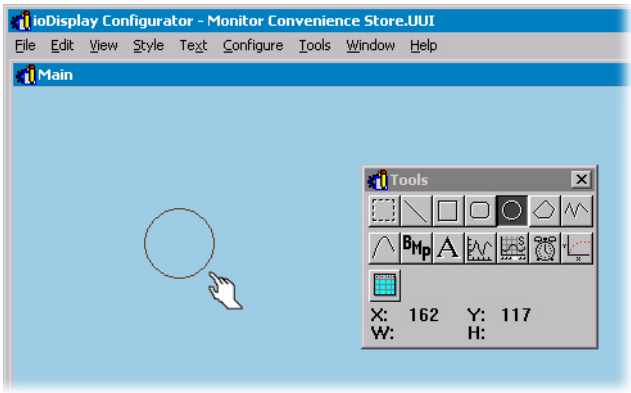
Creating an LED Display

This diagram shows the information flow behind the dynamic LED display you will create in the following steps:



1. Create an LED.

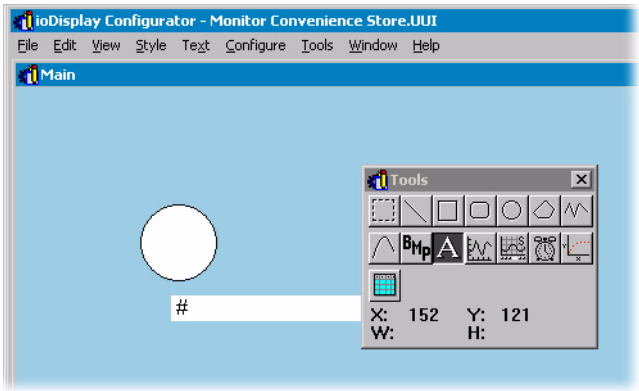
a. Select the Circle tool from the *Tools* window.
b. While holding the CTRL key, click and drag the mouse to create a circle.



The dynamic properties of this graphic will be added later.

2. Create dynamic text.

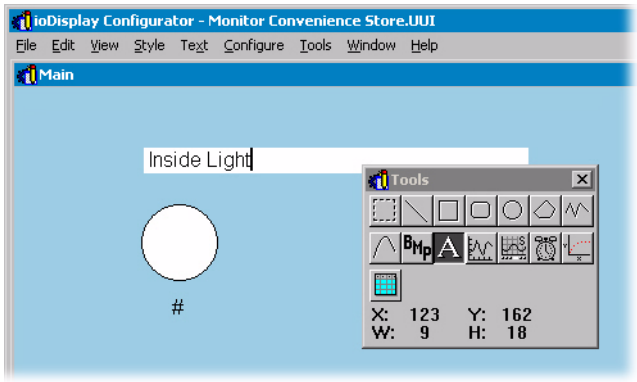
a. Choose the Text tool.
b. Click below the circle and type #.
c. Click below the # to anchor the text.

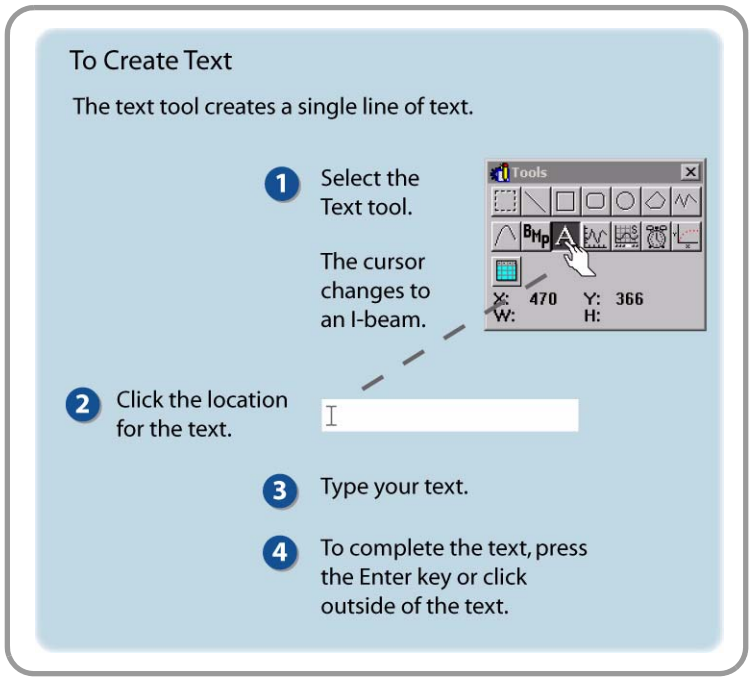


The pound symbol (#) is a special character in ioDisplay that is used within text fields to display values. The dynamic properties of this text field will be added later.

3. Create a text label.

a. With the Text tool still selected, click above the circle.
b. Type `Inside Light` and then click elsewhere to anchor the text.

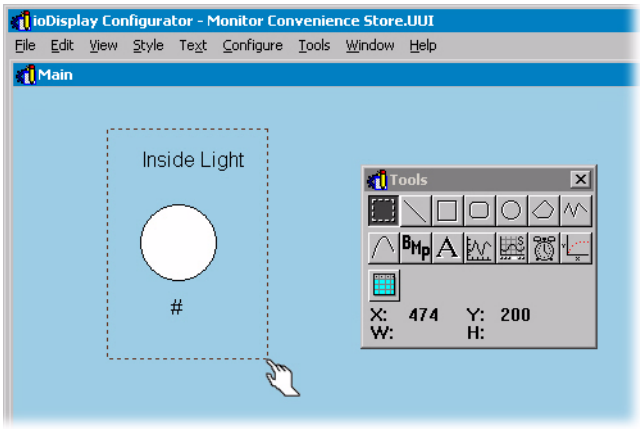




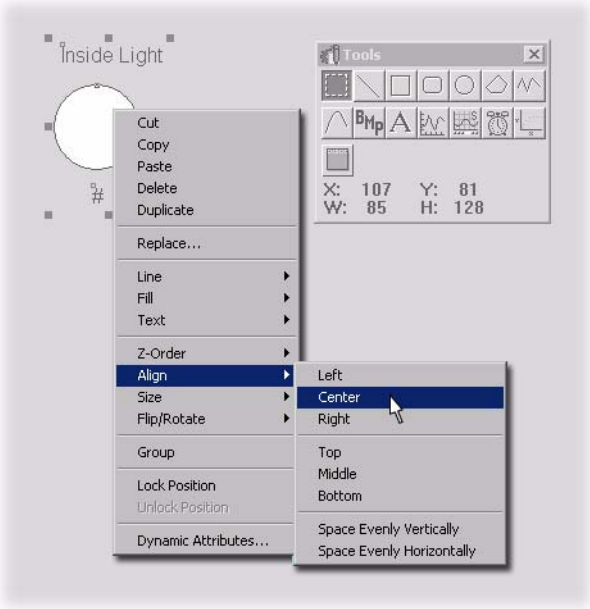
4. Align the objects.

a. Choose the Select tool.

b. Click and drag a rectangle around all three objects to select them.



c. Right-click the items, and from the pop-up menu, select *Align* ⟶ *Center*.



Animating the Light

Once an object is drawn, the next step is to assign attributes for the object. Attributes determine how that object reacts to its linked tag. An object can change appearance or position in response to a change in the tag value. An object can also be used by an operator to send information to a tag.

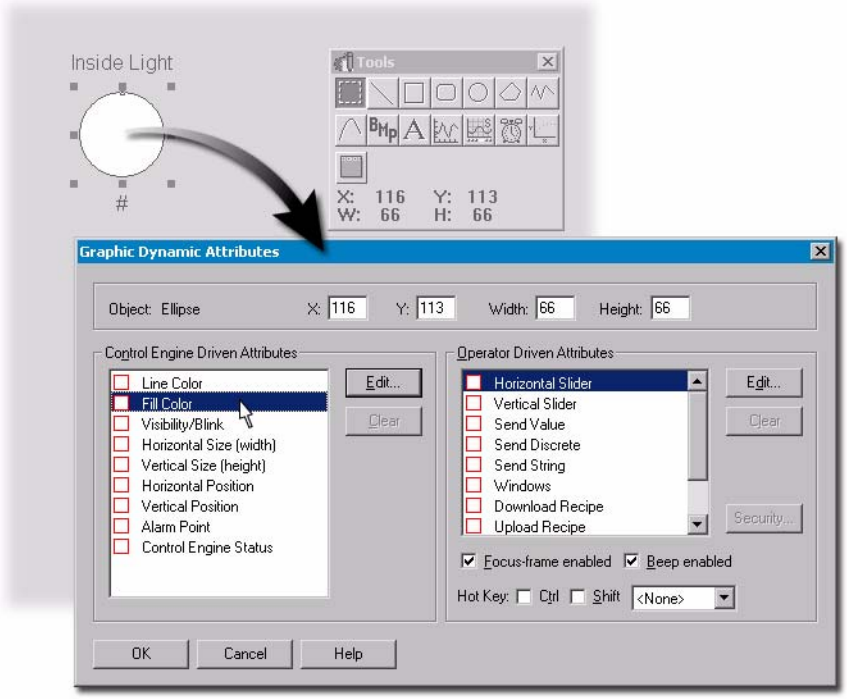
For the first attribute, you will configure the circle to be solid red when the light is on.

1. Click an empty work space to deselect the three objects.

2. Select the graphical dynamic attributes.

a. Double-click the circle.

b. Select *Fill Color* under *Control Engine Driven Attributes*.



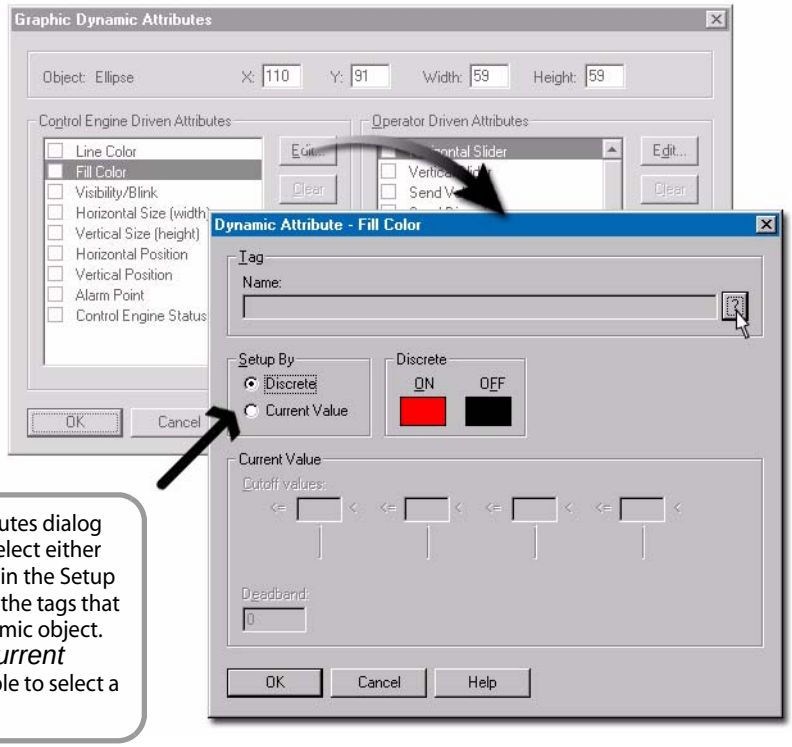
c. Click *Edit*.

3. Select discrete tags.

a. In the Setup By field, select *Discrete*.

b. Click the Tag Selection button (?).

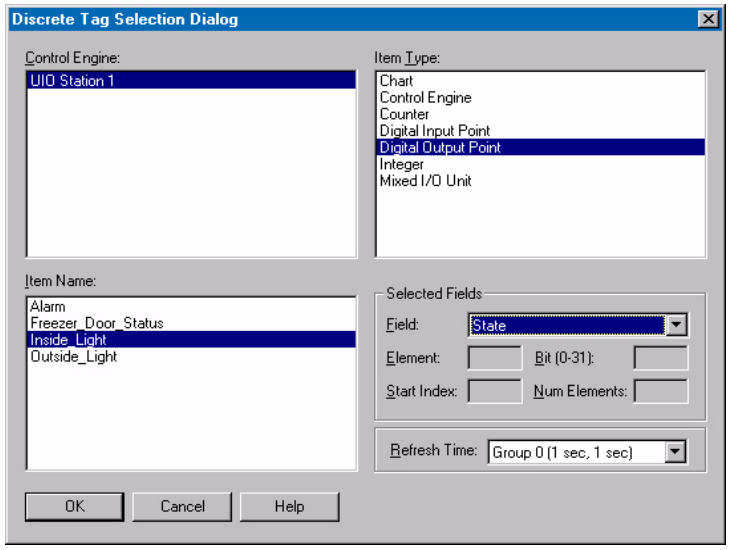
This button opens the Discrete Tag Selection dialog box.



NOTE: In the Dynamic Attributes dialog box, it is very important to select either the Value or Discrete option in the Setup By box, as this choice affects the tags that will be available to this dynamic object. For example, if you chose *Current Value*, you would not be able to select a digital point.

c. In the Item Type field, select *Digital Output Point*.

Notice that all of the digital output points that were configured in your ioControl strategy are listed in the *Item Name* field. Because ioDisplay and ioControl share a database, tag names do not need to be re-entered, merely selected.

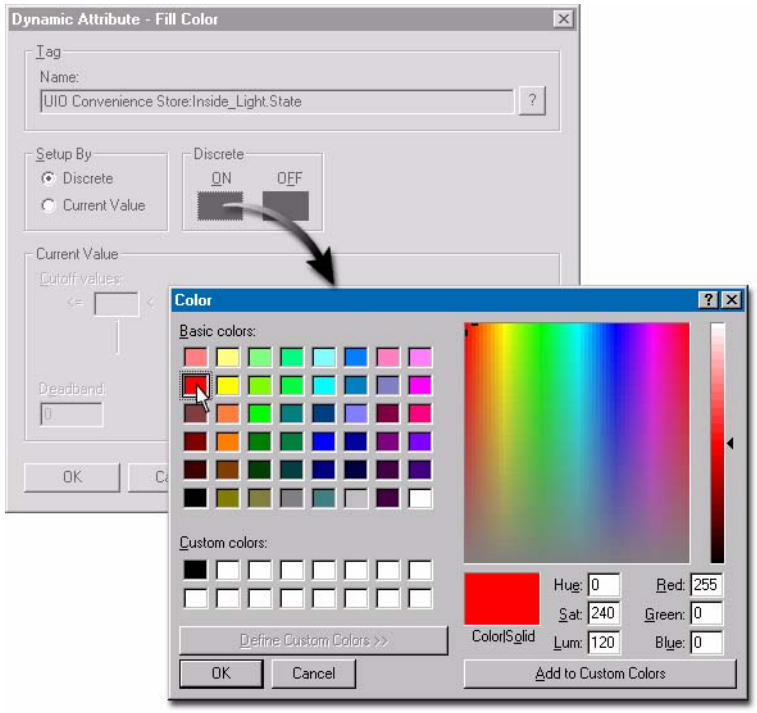


d. In the Item Name field, select *Inside_Light*.

e. Click *OK* to close the Discrete Tag Selection dialog box.

4. Select red fill color for the On state.

a. In the *Discrete* field, click the black box underneath *ON*.



b. Select a bright red, and click *OK* to close the Color dialog box.

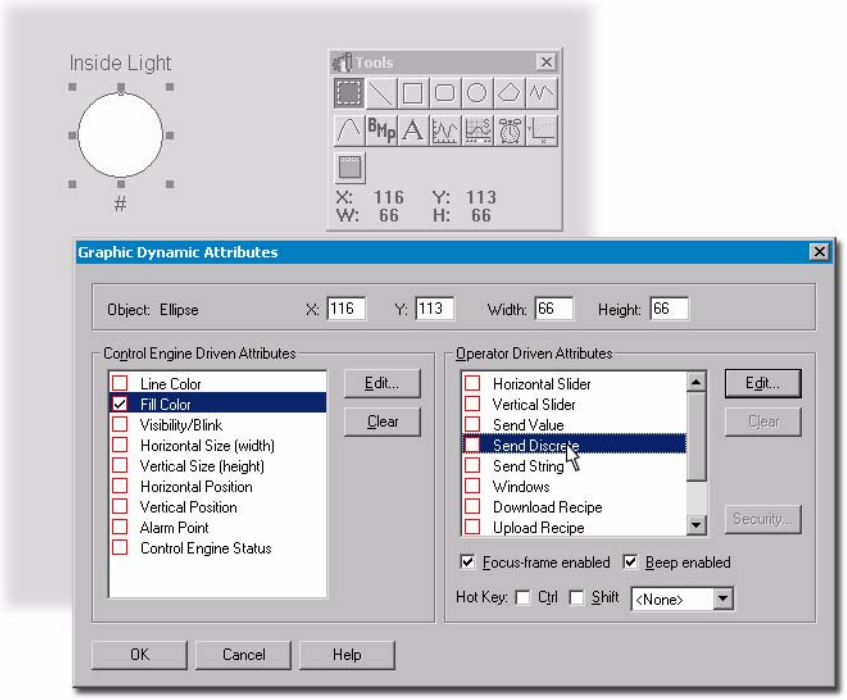
c. Click *OK* to close the Dynamic Attribute dialog box.

Leave the Graphic Dynamic Attributes dialog box open.

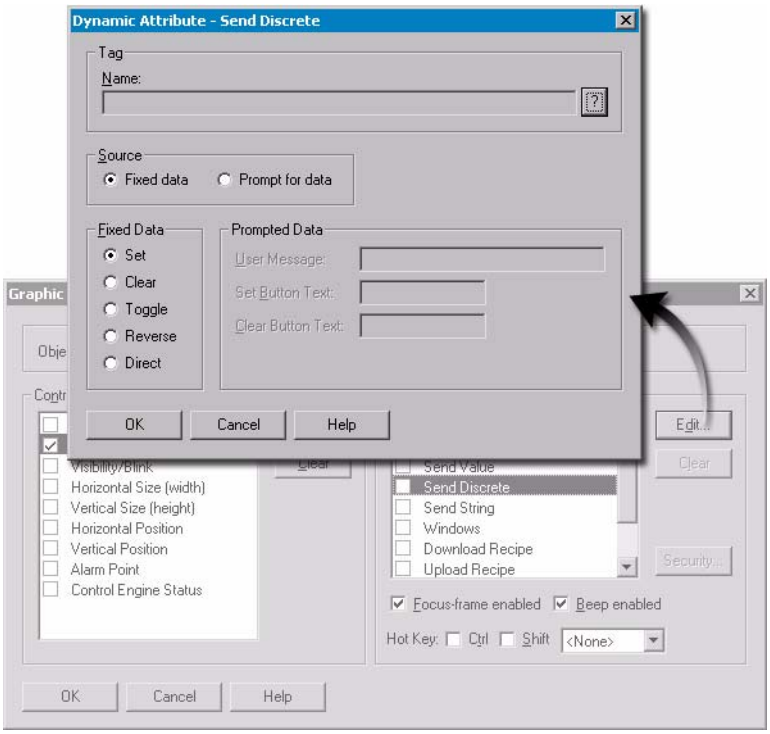
5. Select an operator-driven attribute.

In this step you will add an operator-driven attribute that will turn the light off and on when the operator clicks the circle.

a. From the Graphic Dynamic Attributes dialog box, click *Send Discrete*.



b. Click *Edit* to open the Dynamic Attribute dialog box:

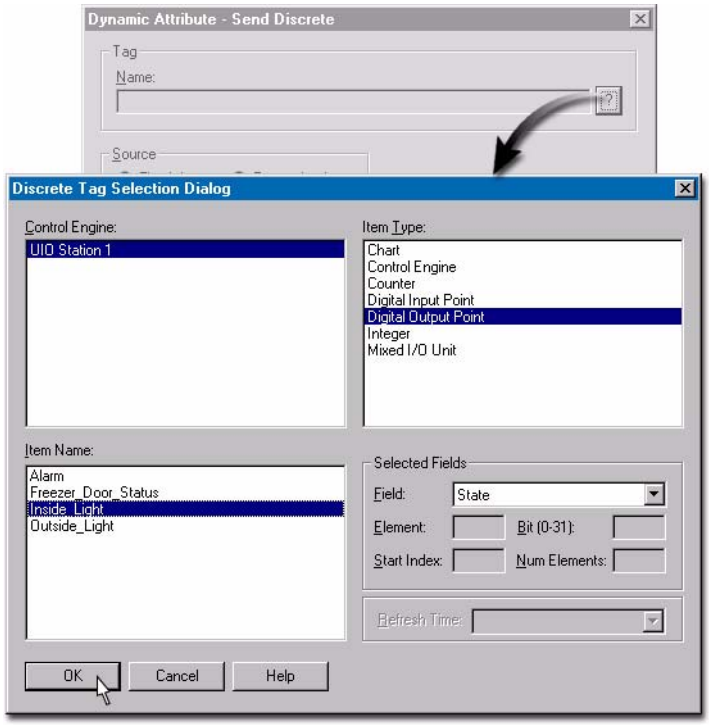


c. Click the Tag Selection button(?).

d. Under Item Type, select *Digital Output Point*.

e. Under Item Name, select *Inside_Light*.

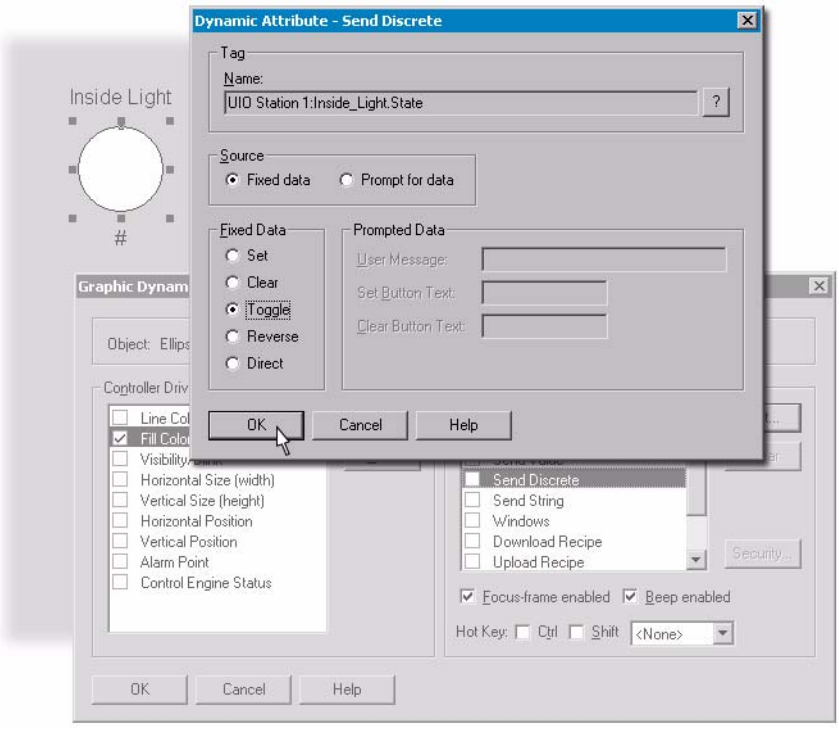
f. Click *OK* to close the Discrete Tag Selection dialog box.



6. Select Toggle action.

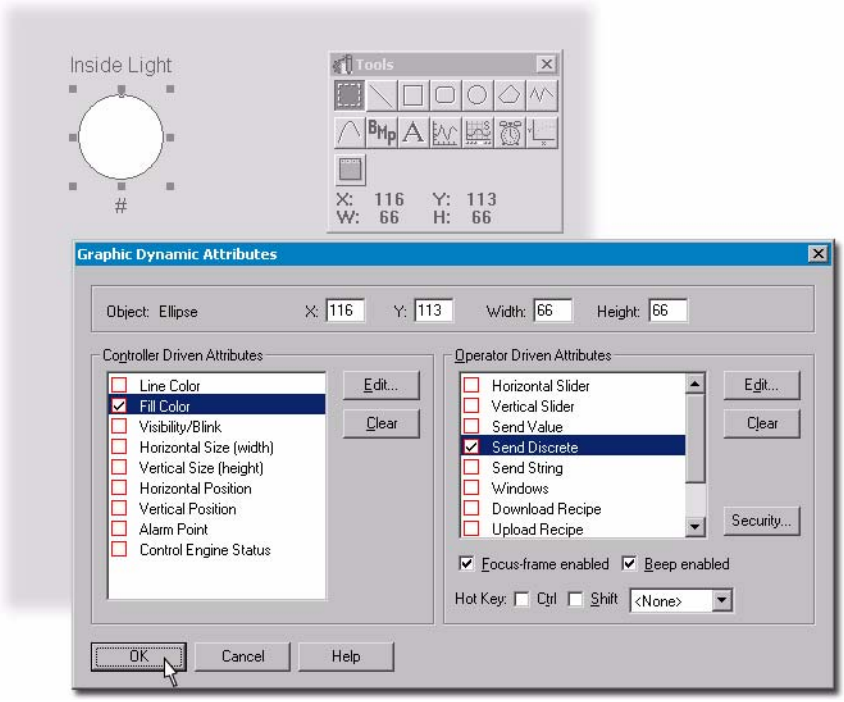
a. Under Fixed Data, select *Toggle*.

b. Click *OK* to close the Dynamic Attribute dialog box.



7. Close the Graphic Dynamic Attributes dialog box.

Click *OK*.

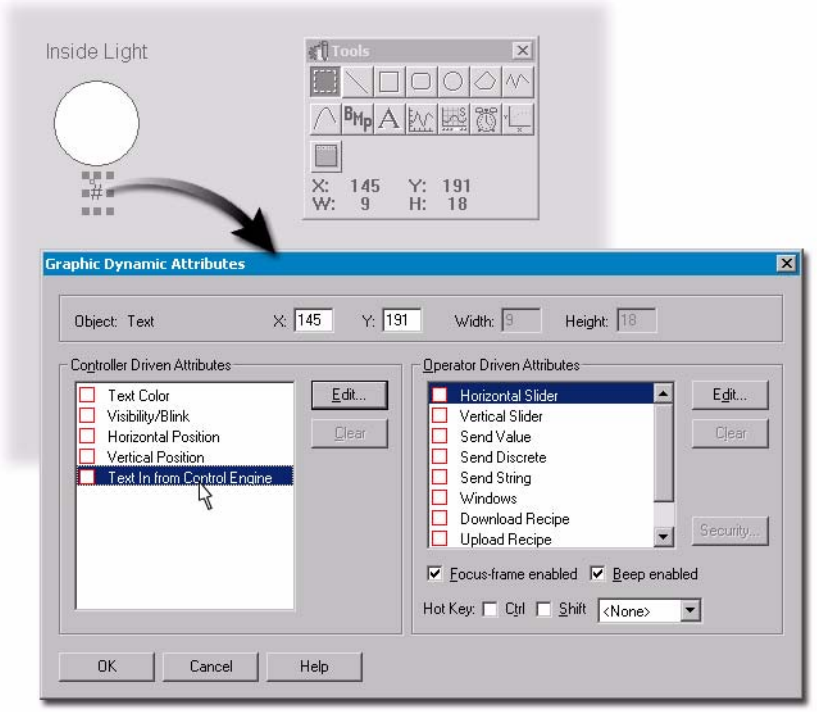


Animating the Text

Next, you will configure the string underneath the light to display On or Off, depending on the state of the light.

1. Open the Graphic Dynamic Attributes dialog box.

a. Double-click the #.

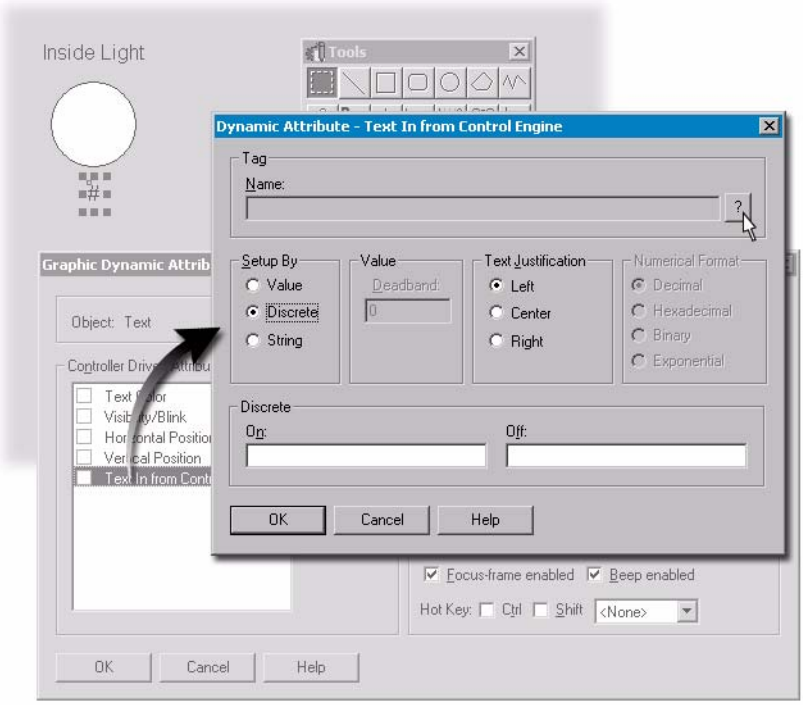


b. Double-click *Text in from Control Engine*.

2. Create a dynamic attribute.

a. In the *Setup By* field, select *Discrete*.

b. Click the *Tag Selection* button.

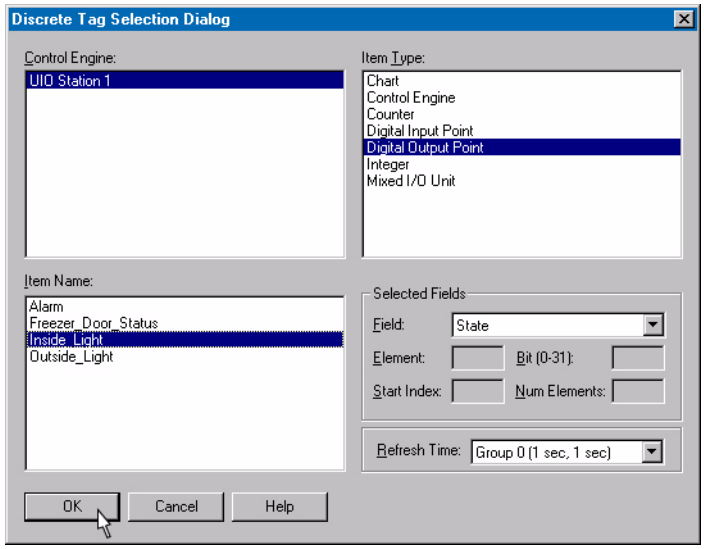


3. Select a Discrete tag.

a. In the *Item Type* field, select *Digital Output Point*.

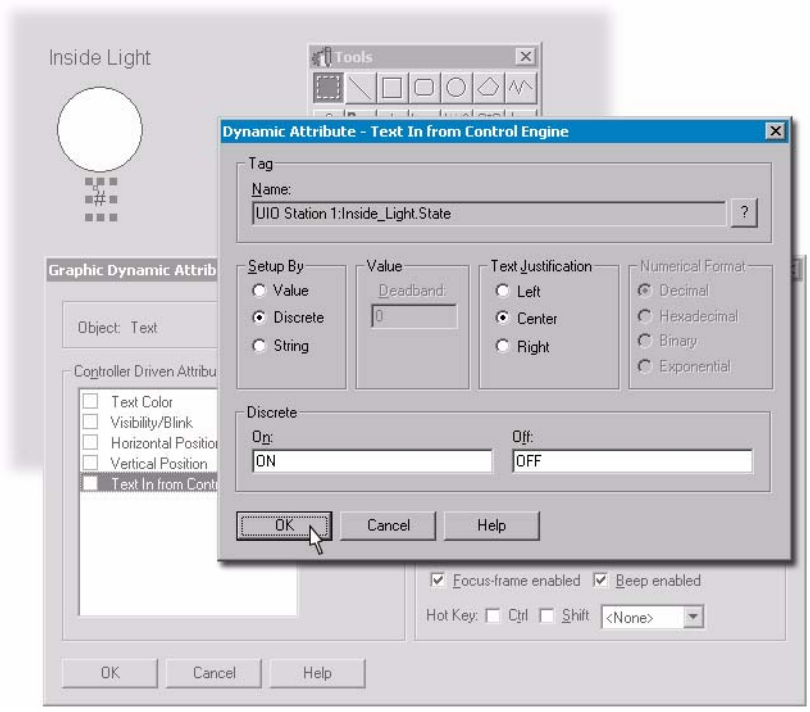
b. In the *Item Name* field, select *Inside_Light*.

c. Click *OK* to close the Discrete Tag Selection dialog box.



4. Complete the dynamic attributes.

a. In the *On* field, type ON.
b. In the *Off* field, type OFF.
c. In the *Text Justification* field, select *Center*.



Note: Any text can be put in the On and Off fields, such as Open and Closed, Running and Stopped.

d. Click *OK* to close the Dynamic Attribute dialog box.
e. Click *OK* to close the Graphic Dynamic Attributes dialog box.

Making a Duplicate LED Display for the Outside Light

1. Select all three objects.

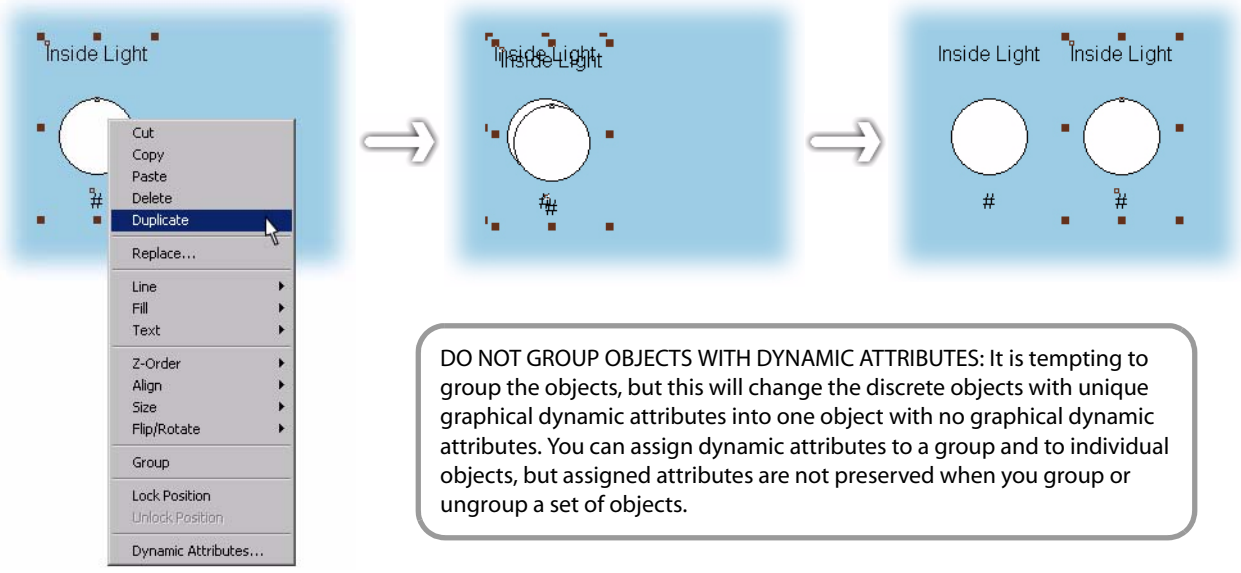
Select the circle and two text fields by doing either of the following:

- Using the Select tool, click and drag a rectangular area embracing the three objects.
- Click the first object and then shift-click the remaining two.

2. Copy and paste.

a. Right-click the selected objects.
b. Choose *Duplicate*.

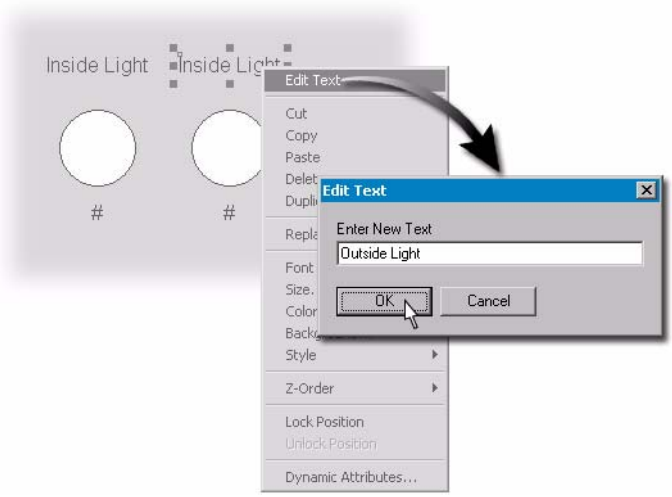
c. Drag the new objects to the right. (Move the tool palette if it is in your way.)



DO NOT GROUP OBJECTS WITH DYNAMIC ATTRIBUTES: It is tempting to group the objects, but this will change the discrete objects with unique graphical dynamic attributes into one object with no graphical dynamic attributes. You can assign dynamic attributes to a group and to individual objects, but assigned attributes are not preserved when you group or ungroup a set of objects.

3. Edit text.

a. Click in the window to deselect the objects.
b. Click the *Inside Light* label and then right-click the *Inside Light* label.
c. Choose *Edit Text*.
d. Type `Outside Light`.
e. Click *OK*.

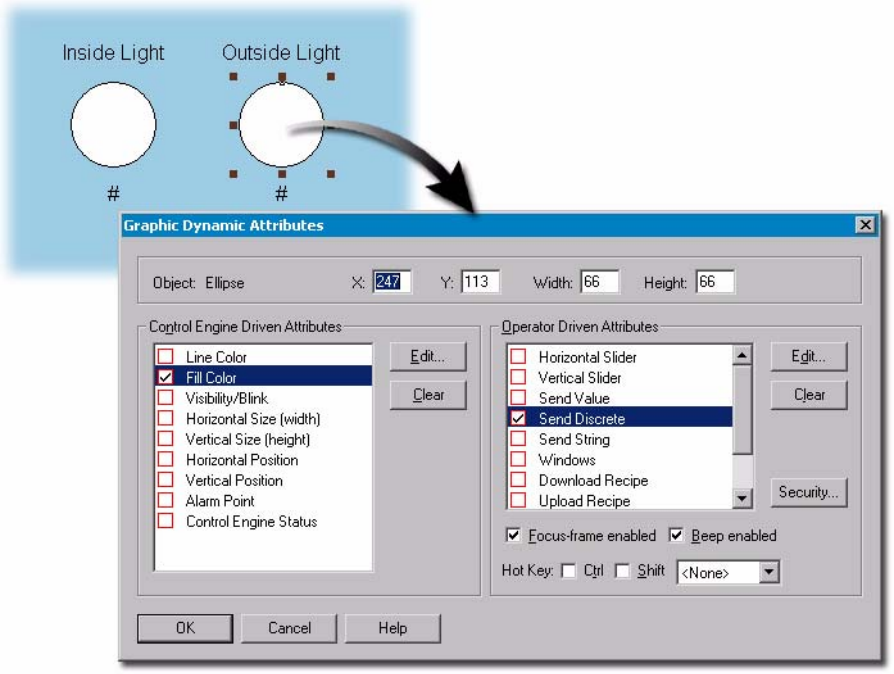


Attaching the Outside Light to Tag

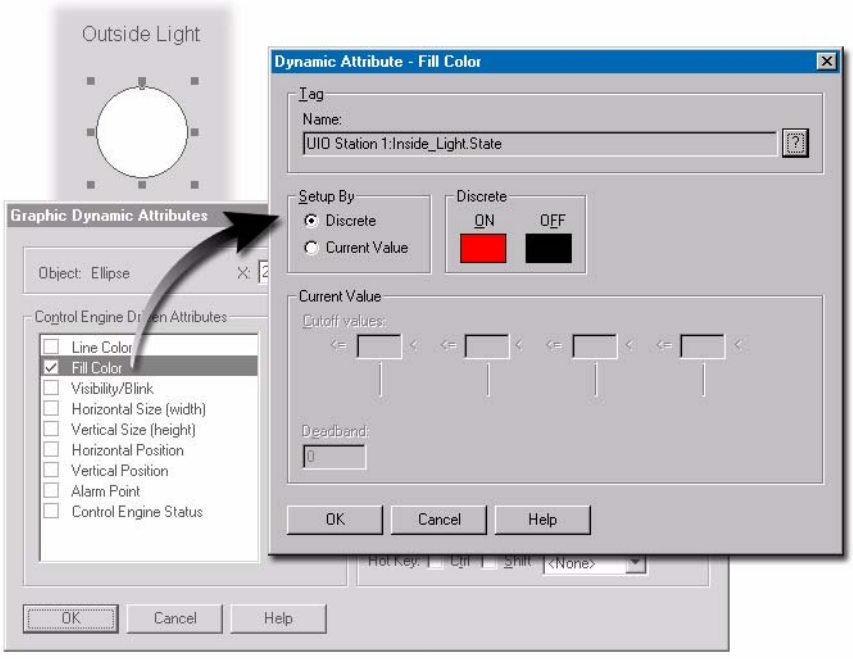
The outside light has the same dynamic attributes as the inside light. You only need to change the tags used by the control engine-driven and operator-driven attributes.

1. Open the Graphic Dynamic Attributes dialog box for the Outside Light graphic.

a. Double-click the outside light circle graphic.

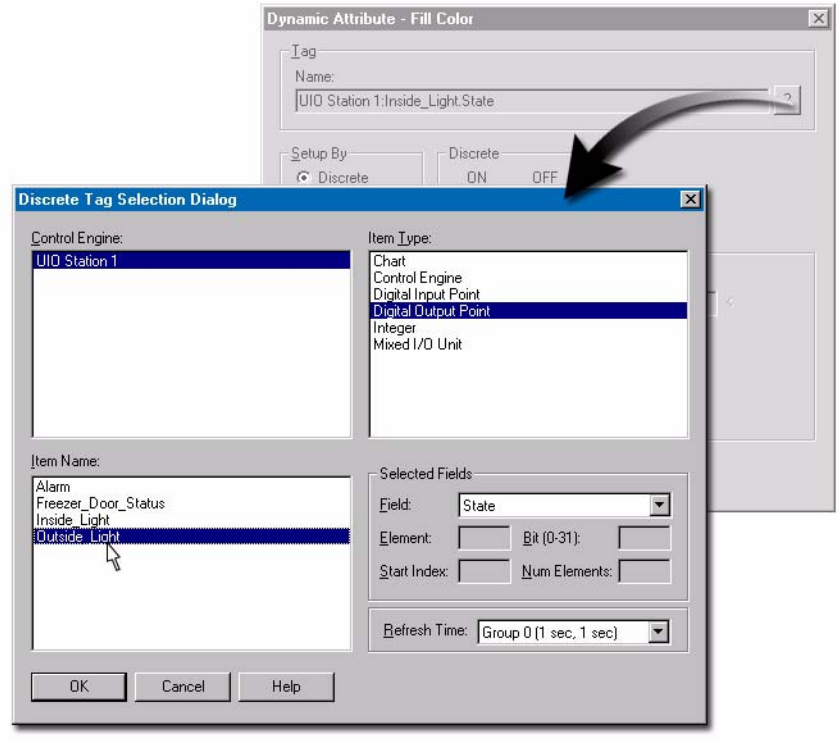


b. Under Control Engine Driven Attributes, double-click *Fill Color*.

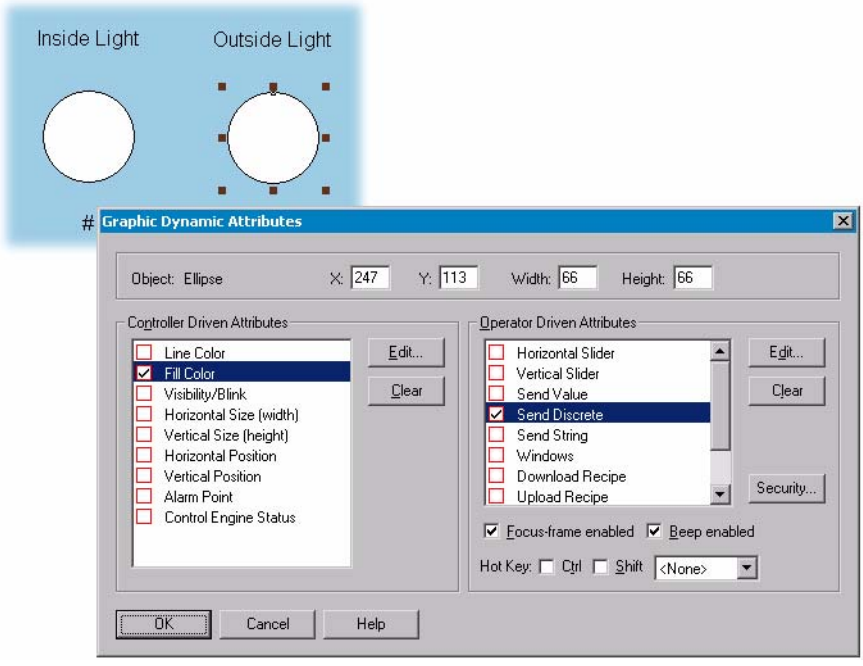


c. In the Dynamic Attribute dialog box, click the *Tag Selection* button (?).

d. Under *Item Name*, select *Outside_Light*.



e. Click *OK* to close the Discrete Tag Selection dialog box.
f. Click *OK* to close the Dynamic Attribute – Fill Color dialog box.

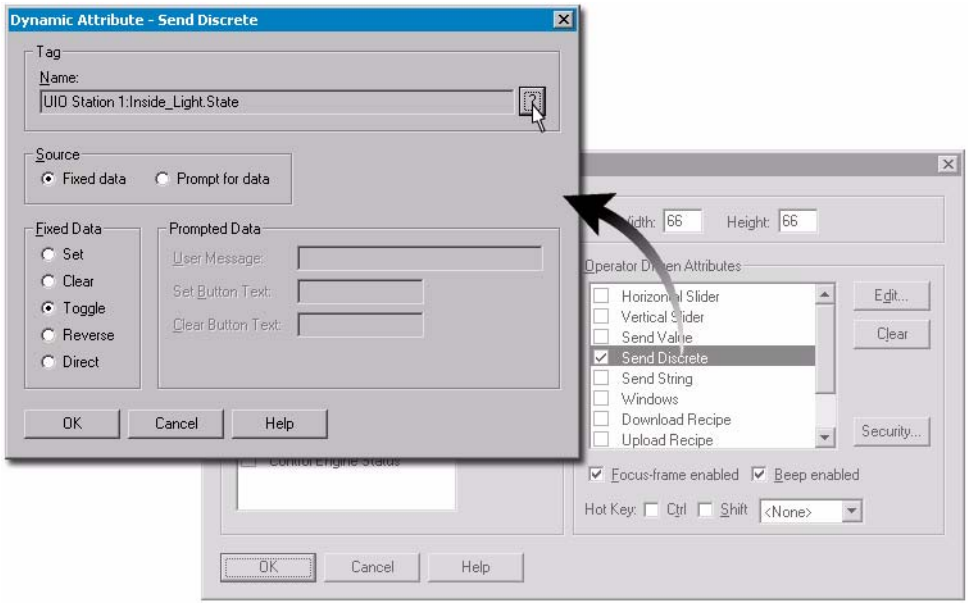


Leave the Graphic Dynamic Attributes dialog box open.

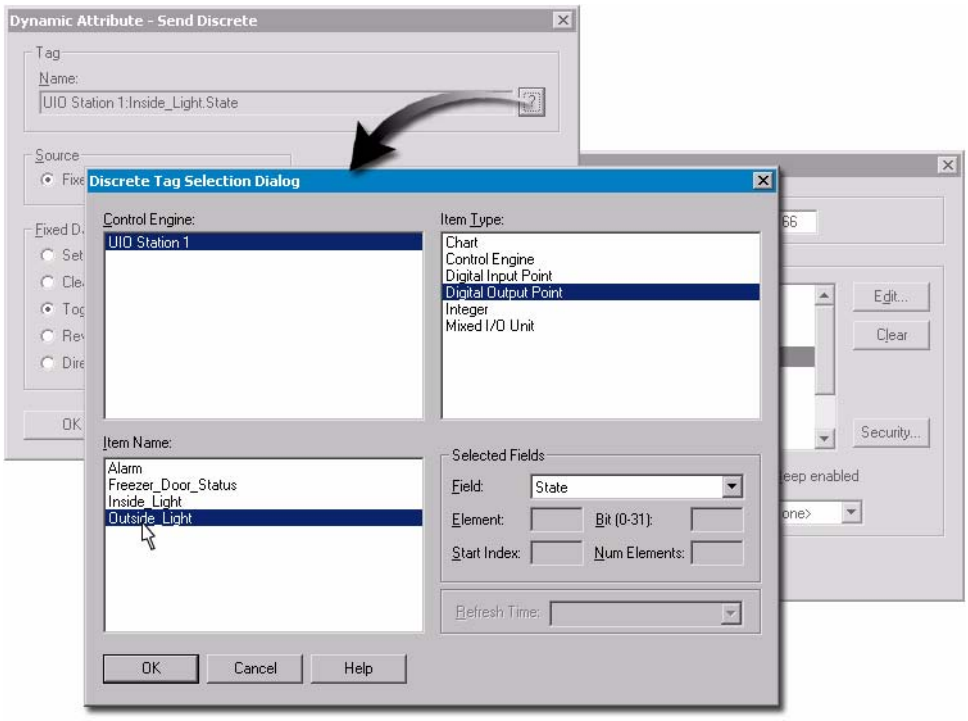
2. Change the Operator-Driven Attribute.

Although we are configuring this graphic with an operator-driven attribute, the digital output point the light is connected to is controlled by the ioControl strategy. If the Digital_Control chart is running, the state of the Photo_Sensor switch, not the operator, will continue to determine whether the outside light is on or off. (You might also choose to delete or disable the operator-driven attribute by simply unchecking the box next to Send Discete. You might also wish to keep this attribute to provide a way of controlling the lights if you stop the ioControl strategy.)

a. In the Graphic Dynamic Attributes dialog box, double-click the *Send Discrete* attribute.



b. In the Dynamic Attribute dialog box, click the *Tag Selection* button.

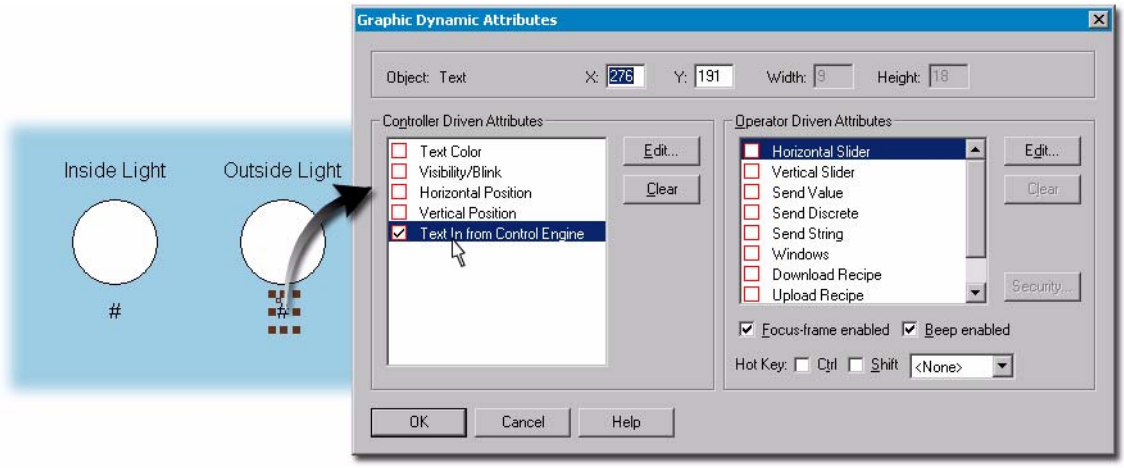


c. Select *Outside_Light*.

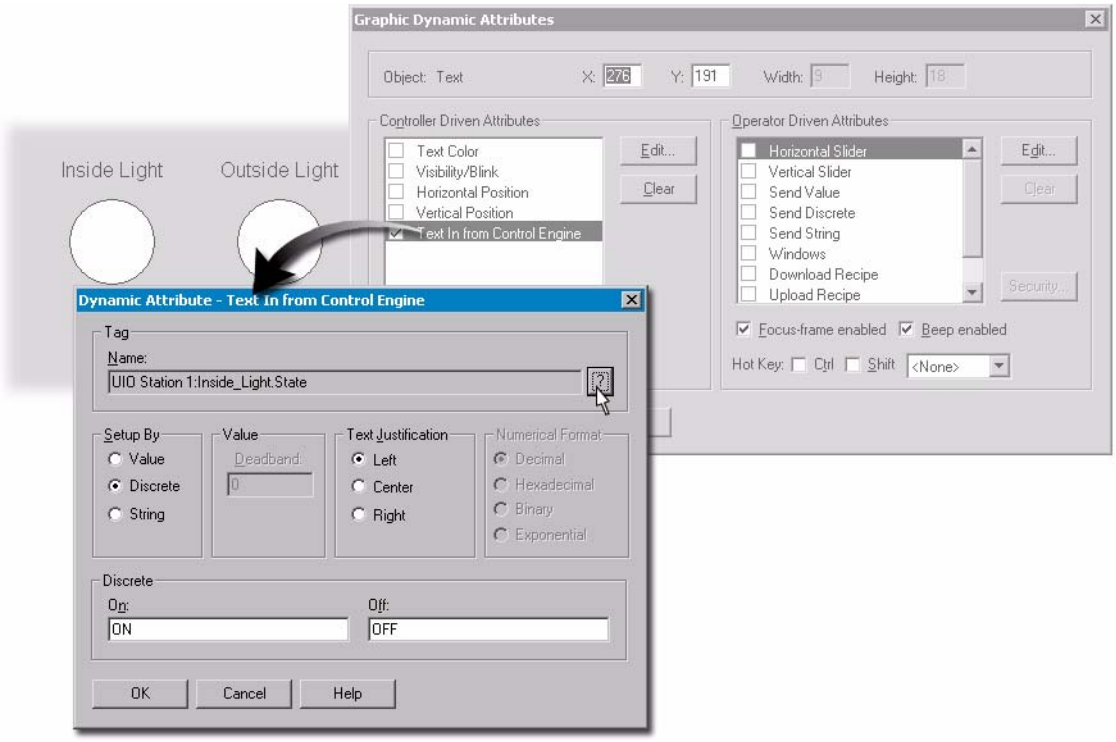
d. Close all dialog boxes.

3. Attach outside light text to tag.

a. Double-click the outside light's dynamic text field.

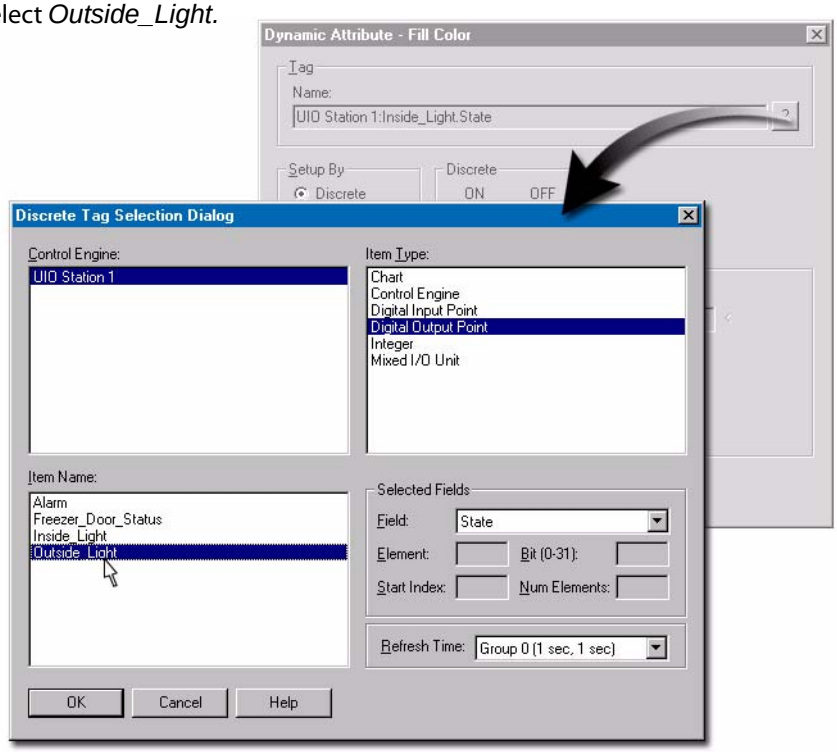


b. In the Graphic Dynamic Attributes dialog box, double-click *Text in from Control Engine*.



c. In the Dynamic Attribute dialog box, click the *Tag Selection* button.

d. Select *Outside_Light*.



e. Close all dialog boxes.

Testing the Project

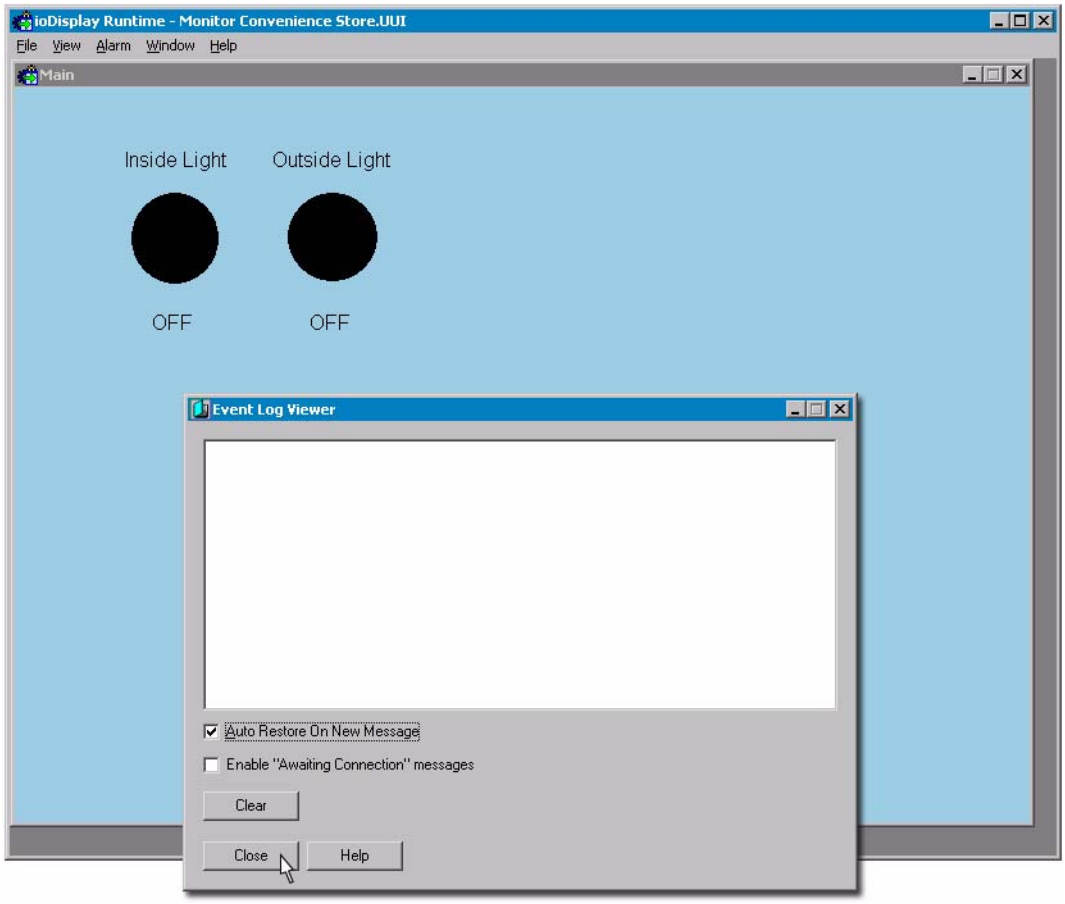
1. Run your project.

Select *File* ➞ *Save Project and Load Runtime*.

2. Close the Event Log Viewer.

Your ioDisplay project is automatically loaded and the Event Log Viewer window appears. Use this dialog box to confirm a successful connection to your control engine.

Click *Close* to close the Event Log Viewer.



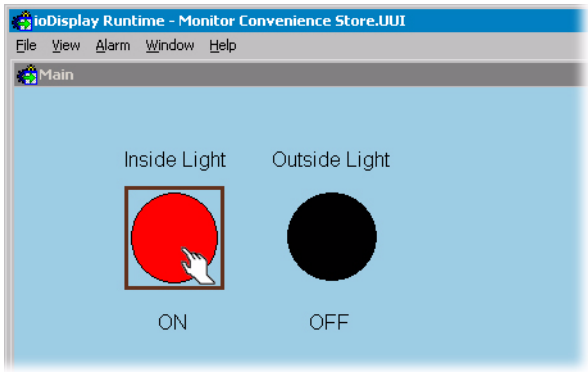
3. Use the operator interface.

a. Turn on and off the Photo Sensor switch on the load panel.

The change in state of the outside light (triggered by the Photo Sensor switch) shows red and black and ON and OFF.

b. Click the Inside Light LED in your project window.

This will turn off the Inside Light on your load panel.



c. Click the Inside Light LED again, and the Inside Light turns off.
d. Click the Outside Light LED in your project window.

   You will *not* be able to change the state of the light for very long. (Remember: ioControl is controlling this light based on the state of the photo sensor.)

4. Close ioDisplay Runtime.

a. Choose *File* ➞ *Exit ioDisplay Runtime*.
b. Click *Yes* to acknowledge the Exit ioDisplay Runtime message.

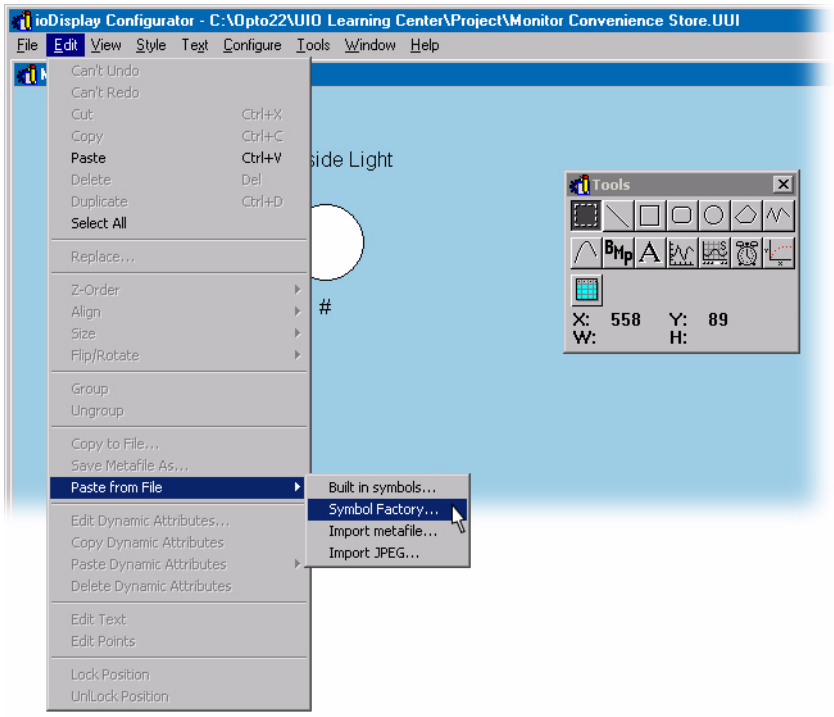
Creating a Tank Level Display

You will create a tank graphic with a bar graph that moves up and down proportionally to reflect the level of the tank.

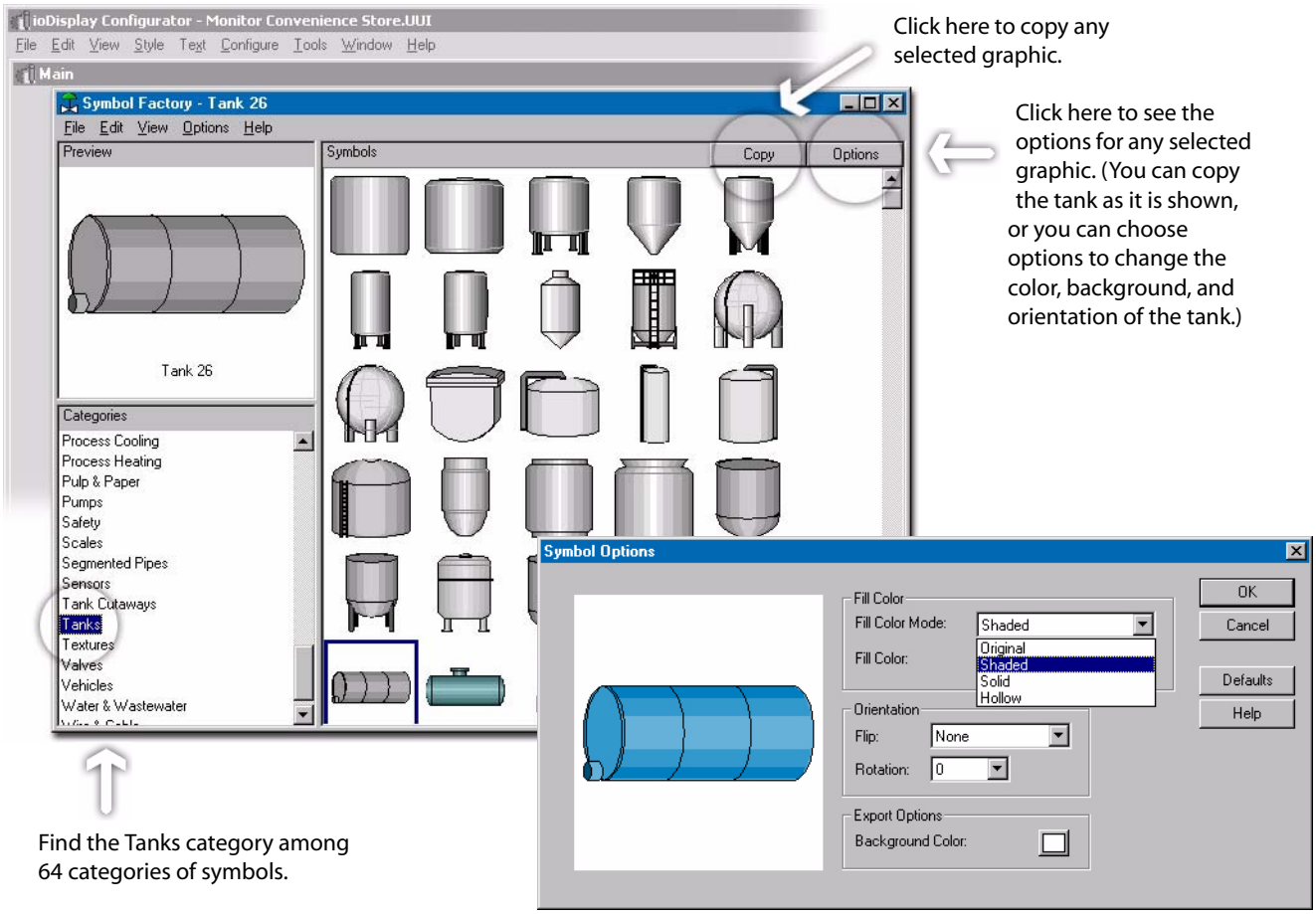
1. Locate the tank graphic from Symbol Factory.

First you will import a picture of a tank picture from the Symbol Factory, which is a graphics library built into ioDisplay. The Symbol Factory is a library of over 3,000 symbols for industrial automation, including pumps, pipes, valves, tanks, mixers, motors, ducts, electrical symbols, flow meters, material handling, sensors, PLCs, transmitters, and ISA symbols.

a. From the *Edit* menu, select *Paste from File* ⟶ *Symbol Factory.*



b. Select *Tanks* in the *Categories* field.



2. Change the options for a Symbol Factory graphic.

a. Click a tank to select it.
b. Click *Options*.
c. Select *Shaded* from the *Fill Color Mode* menu.
d. Click the *Fill Color* box and choose a color.
e. Click the *Defaults* button if you wish to restore the image.
f. Click *OK* to close the Symbol Options dialog box.

3. Paste the Symbol Factory graphic into ioDisplay.

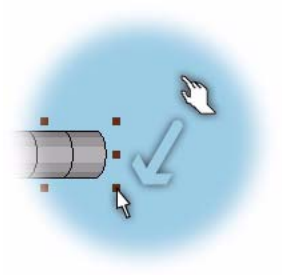
a. Make sure the tank you want to paste into ioDisplay is selected.
b. Click *Copy*.
c. Close the Symbol Factory by clicking the X in the upper right-hand corner.
d. Click anywhere in the Main drawing window to make ioDisplay the active application.
e. Right-click in the Main window and select *Paste*.

Note: Symbol Factory will copy graphics as either a bitmap or a WMF file. To change the file type—for example, to remove the square border of a bitmap—in Symbol Factory, see *Options ➞ Configure*.

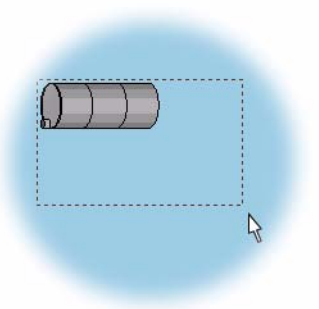
4. Resize the tank graphic.

a. Choose the *Select* tool.
b. Click and drag the tank to the lower left-hand side of the window.
c. Resize the tank by moving your cursor over one of the size handles, clicking and dragging the mouse. Release the mouse when the tank is the desired size.

Resizing a Graphic:



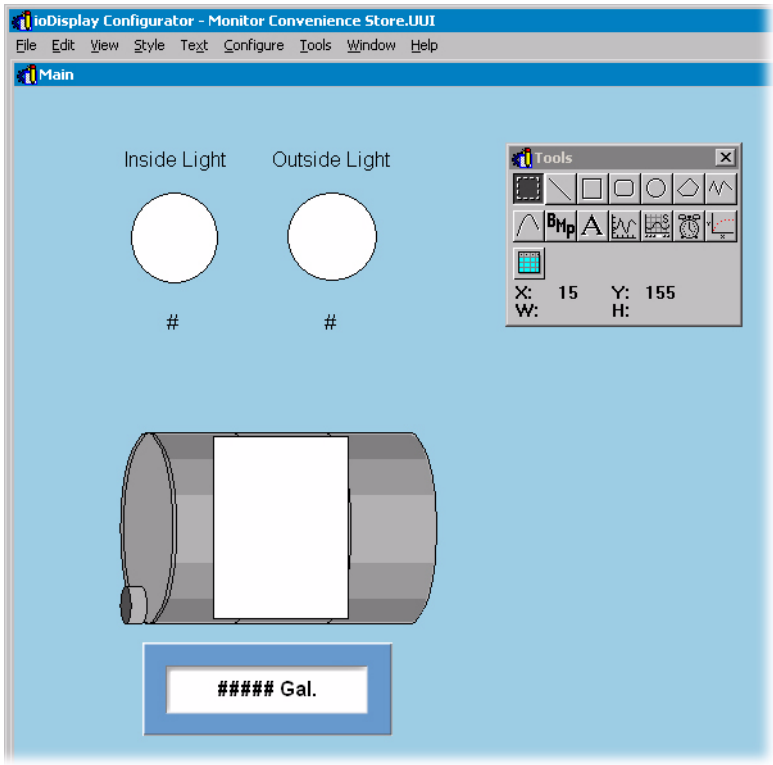
Move the hand cursor to the lower-right drag handle (one of eight small boxes indicating selection).



When the cursor changes to an arrow, click the handle and drag to resize the graphic.

5. Create a dynamic rectangle.

a. Select the Rectangle tool from the Tools window.
b. Position the mouse over the tank.
c. Click and hold the left mouse button.
d. While holding the mouse button, move the mouse in a diagonal motion to create a rectangle.
e. When the rectangle has the desired shape, release the mouse button.



6. Adjust size and placement.

a. Choose the Select tool.

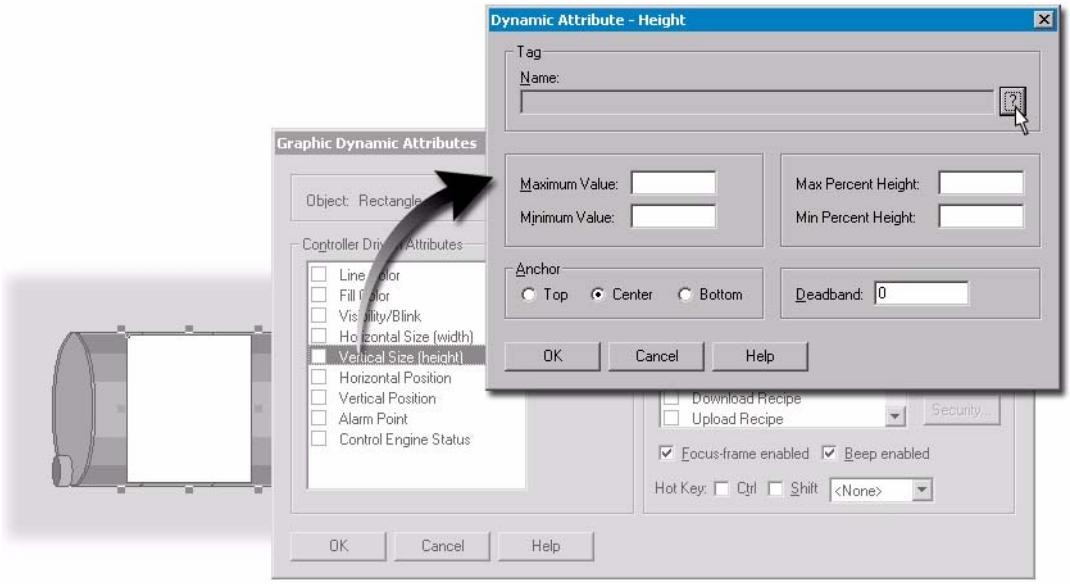
b. If necessary, resize the rectangle or center it on the tank. You can also use the arrow keys on your keyboard to move a selected object.

Assigning Dynamic Attributes

1. Add a vertical size graphic dynamic attribute.

a. Double-click the rectangle.

b. In the Graphic Dynamic Attributes dialog box, double-click *Vertical Size (height)* as the attribute.



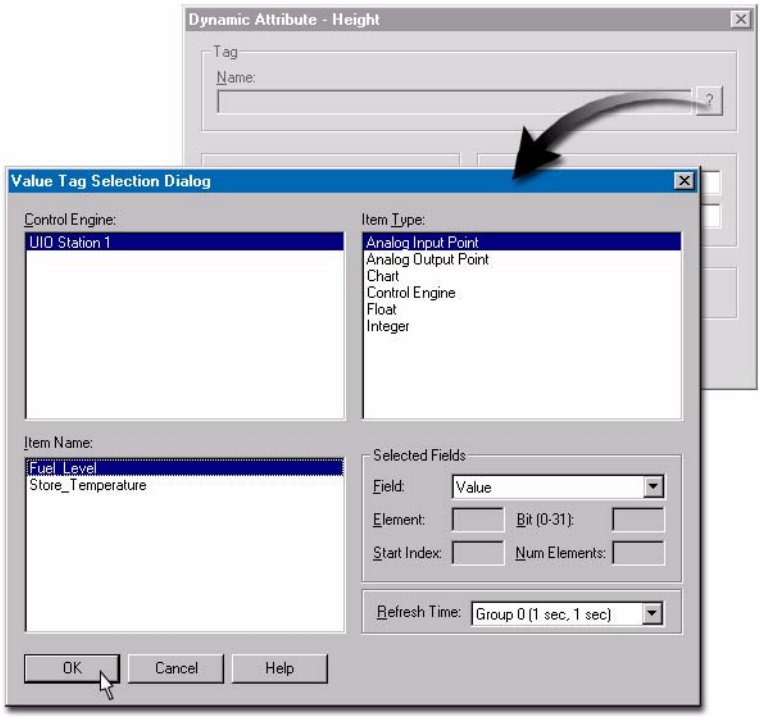
2. Assign a Tag.

a. Click the *Tag Selection* button.

b. In the *Item Type* list, select *Analog Input Point.*

c. In the *Item Name* list, select *Fuel_Level.*

d. Click *OK* to close the Value Tag Selection dialog box.

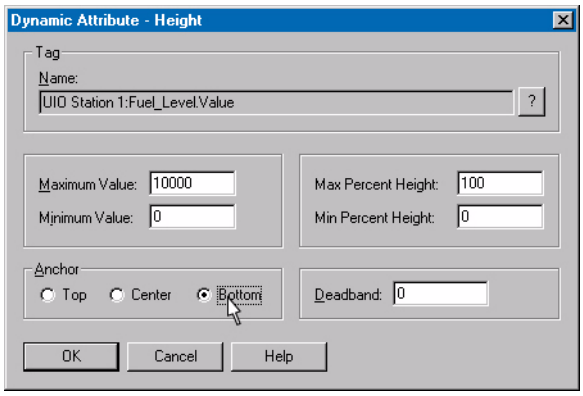


3. Enter the height scaling.

a. Type `10000` in the *Maximum Value* field.

b. Type `0` in the *Minimum Value* field.

You have chosen the full range of values to monitor for this tag (0–10,000). An object does not have to monitor the entire range of values. A subset of values can be monitored.



c. Type `100` in the *Max Percent Height* field.

This field is used to specify the maximum percent height size that the graphic may achieve relative to its size in the Configurator. This percentage is achieved when the tag (Fuel_Level) returns data that is greater than or equal to the Maximum Value.

d. Type `0` in the *Min Percent Height* field.

This field specifies that the object will be 0 percent of its current size when *Fuel_Level* returns a value less than or equal to the *Minimum Value* (in this case, 0).

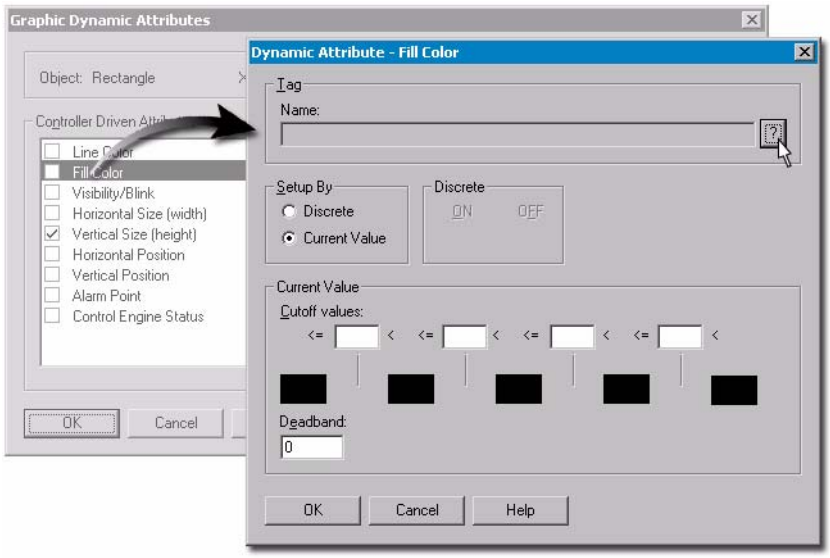
e. Select the *Bottom* option in the *Anchor* field.

The bottom edge of the graphic remains fixed as the object changes size.

f. Click *OK* to close the Dynamic Atribute–Height dialog box.
(Leave the Graphic Dynamic Attributes dialog box open.)

4. Add the fill color graphic dynamic attribute.

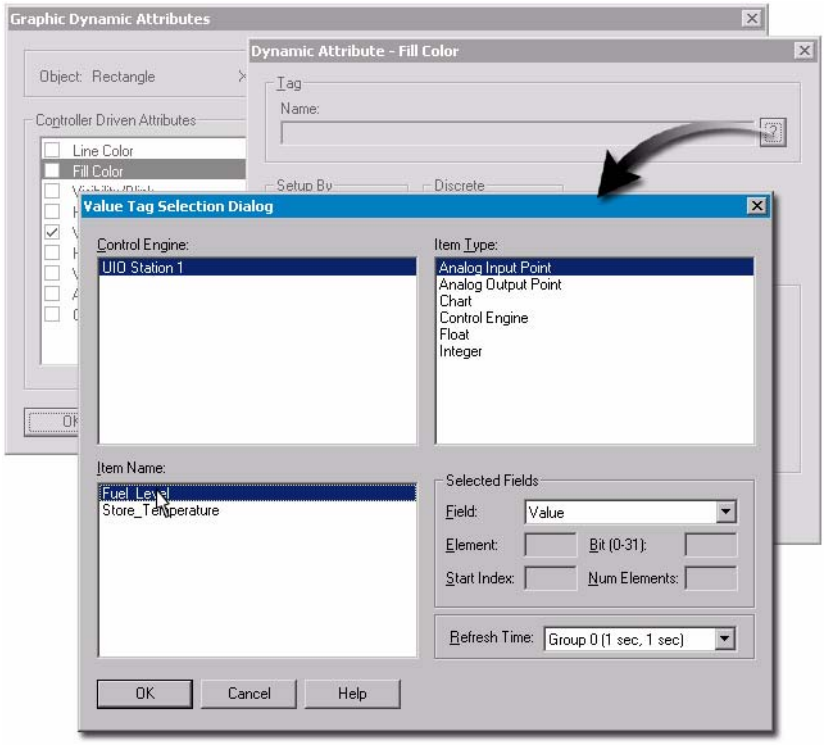
a. Double-click the Fill Color attribute to open the Dynamic Attribute–Fill Color dialog box.



b. Make sure that *Current Value* is selected under Setup By.

5. Assign a tag.

a. Click the Tag Selection button.



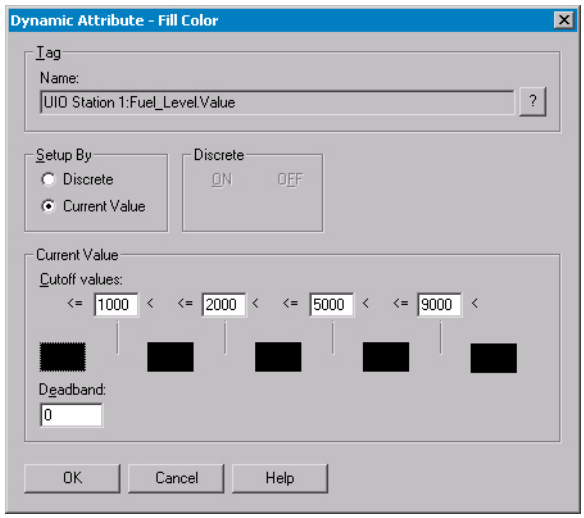
b. In the *Item Type* list, select *Analog Input Point.*
c. In the *Item Name* list, select *Fuel_Level*.
d. Click *OK*.

6. Configure a color change.

You can configure the Fill Color of the tank to be based on I/O values. For example, if the tank level is less than 1000 gallons, the level indicator could change to red.

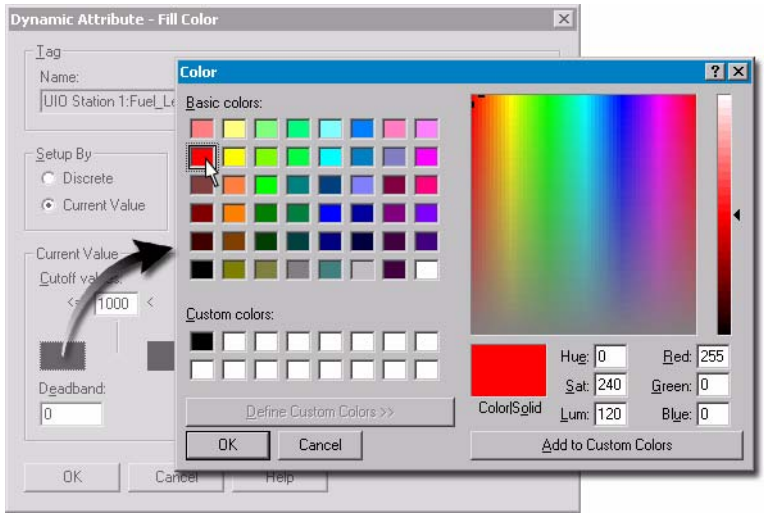
a. In the Dynamic Attribute–Fill Color dialog box, type the following cutoff values, left to right in the order shown here:

```
1000      2000      5000    9000
```



7. Assign colors to each range.

a. Click the left color box (which corresponds to the lowest range).
b. In the Color dialog box, click a red color.



c. Click *OK* to close the Color dialog box.
d. Repeat steps a through c for the remaining ranges, assigning the following colors:

   1000 < 2000: Yellow

   2000 < 5000: Green

   5000 < 9000: Blue

   9000 <: Blue

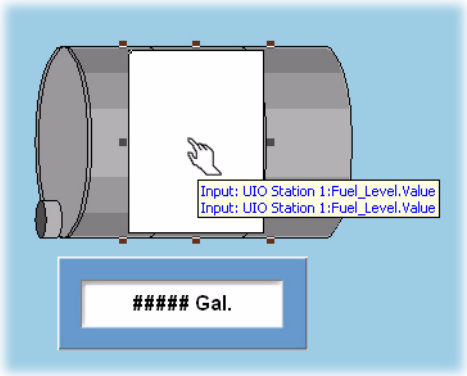
e. When you've assigned colors to all five ranges, close all dialog boxes.

8. Verify the tags assigned to your graphic.

You can place the selection cursor over a graphic to read the tags assigned to it.

a. Click the rectangle.

b. Place the cursor over the rectangle.



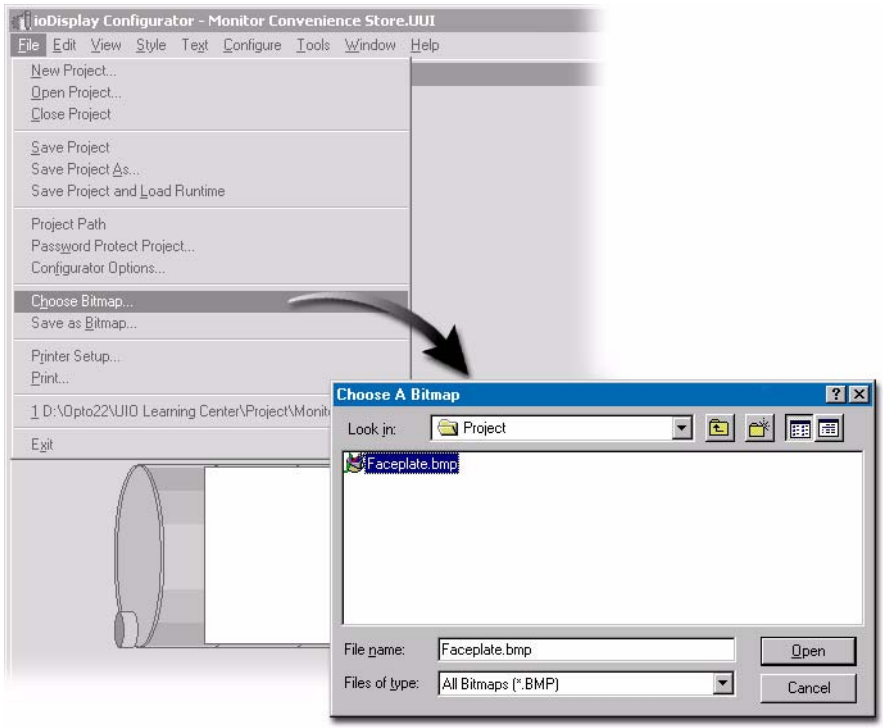
Both tags should be as shown here.

Using Imported Bitmaps in a Graphical Display

In this section, you will bring in a bitmap to use as a display. The built-in graphic support includes a selection of *.bmp (bitmap) files. You can also use a bitmap created in other programs. In this example, we created a faceplate bitmap for you to use.

1. Load the bitmap.

a. Select *Choose Bitmap* from the *File* menu.



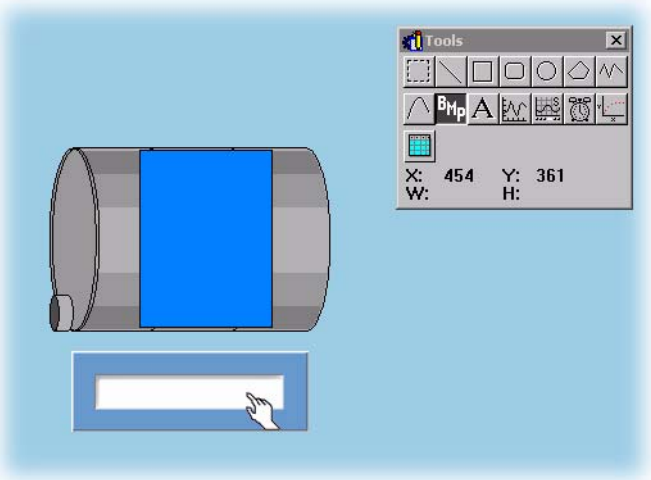
b. Navigate to the **C:\Program Files\Opto22\UIO Learning Center\Project** directory.

c. Double-click **Faceplate.bmp**.

The bitmap graphic Faceplate.bmp is now loaded. Every time you use the Bitmap tool on the Tools Palette, you will place this graphic until you repeat these steps to load a different graphic.

2. Place a bitmap graphic.

a. Choose the Bitmap tool from the *Tools* window.



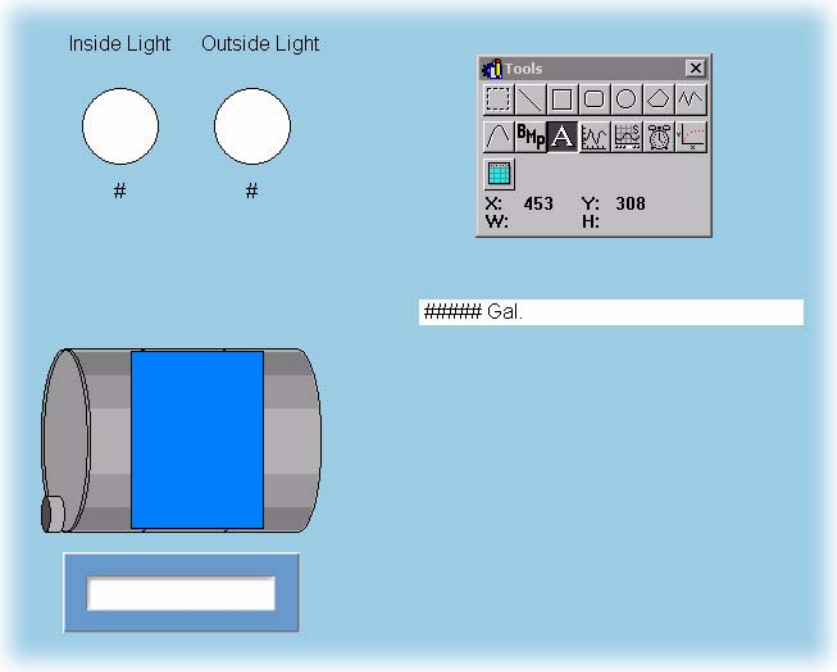
b. Click below the bar graph to place the bitmap.
c. If you need to move the bitmap, click the *Select* tool, and click and drag the bitmap to the desired location.

NOTE: You can also move a selected object using the arrow keys.

3. Create text label.

a. Select the Text tool from the *Tools* window.
b. Click anywhere in the drawing window.

c. In the text box, type `##### Gal.`

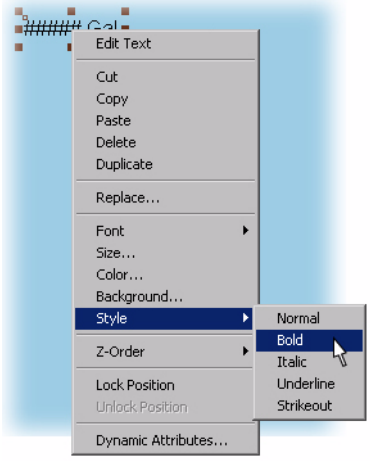


d. Click elsewhere in the drawing window to complete the entry of the text.

You could just type one #. However, by typing five characters (one for each digit in the maximum tank size) you can adjust the font and spacing so that the tank level reading will fit in the display box throughout the entire range. The # character also allows you to format a decimal point. For example, if you want two decimal places to be shown, you would type #####.##.

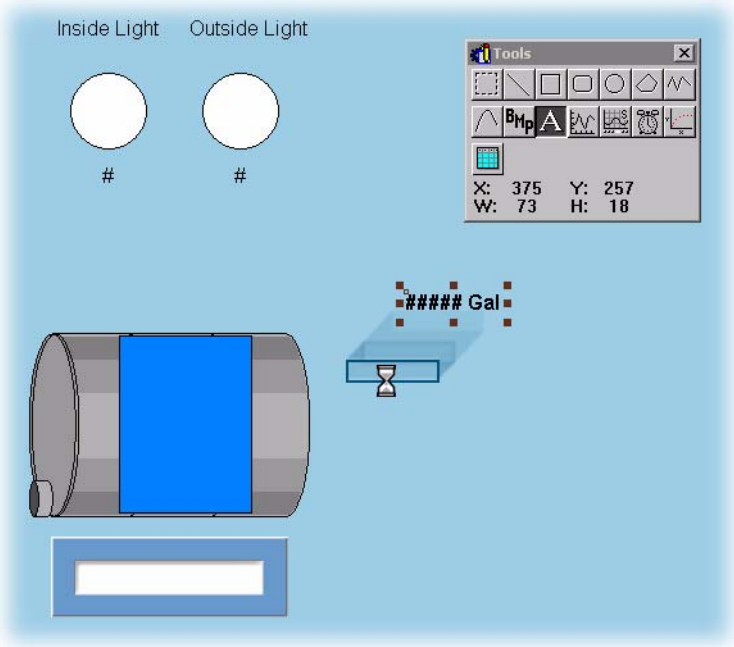
4. Make the text bold.

a. Choose the Select tool and select the text.
b. Right-click the text.
c. Select *Bold* from the Style menu.



5. Drag the text to the display.

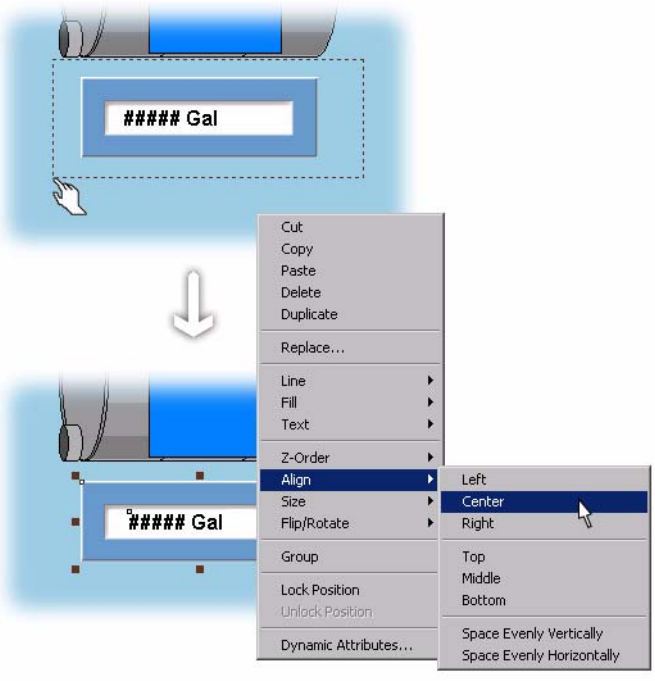
a. Place the mouse over the text.
b. Click and hold the left mouse button.
c. Drag the mouse to move the text to the display bitmap.
d. Release the mouse button when the text is in the desired place.



6. Align the display and text.

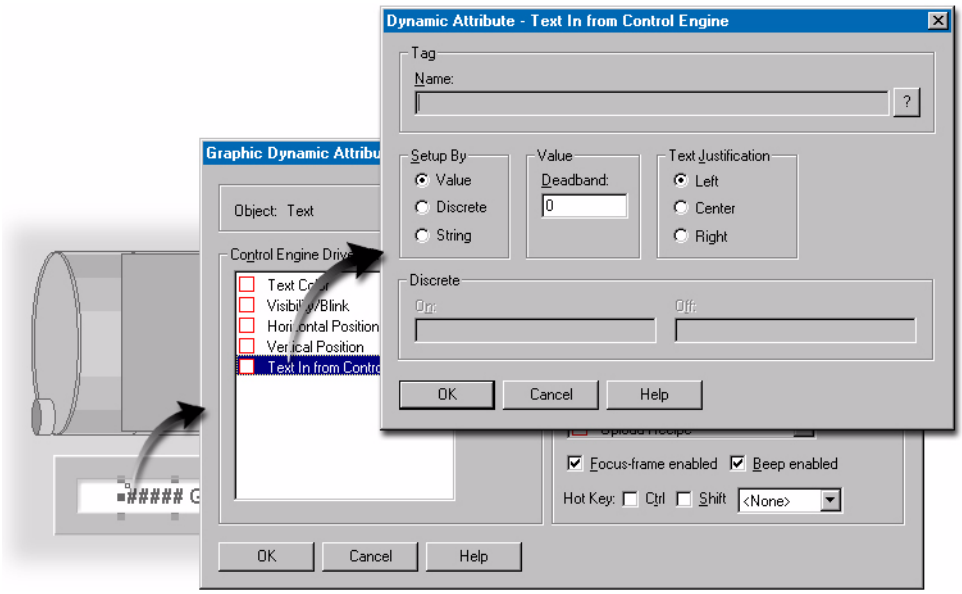
a. Click elsewhere in the window to release any selected objects.
b. Select the text and bitmap.
c. Right-click the selected objects.
d. Select *Align* and then *Center* from the pop-up menu.
e. Right-click the selected objects again to open the same pop-up menu.
f. Select *Align* and then *Middle*.

   This selection aligns the selected graphics along a middle horizontal axis.



7. Assign a tag to the tank level display.

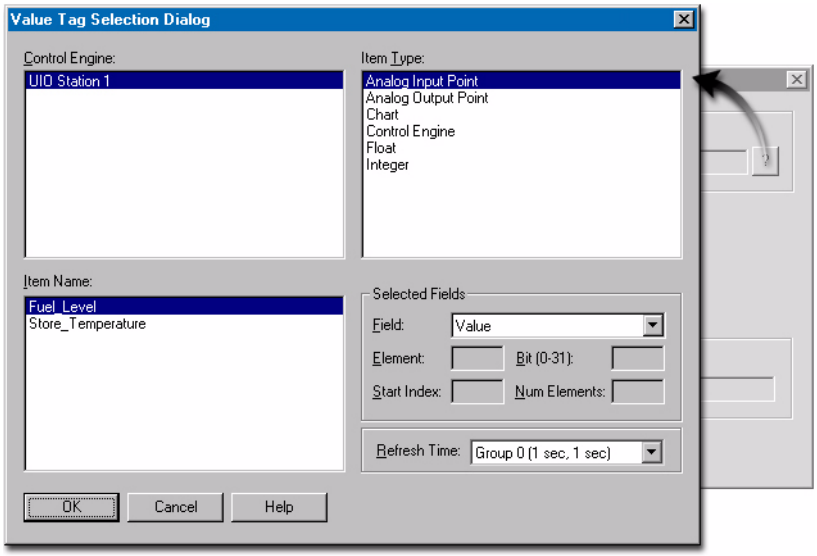
a. Click elsewhere in the drawing window to deselect the bitmap and the text.
b. Double-click the text to open the Graphic Dynamic Attributes dialog box.
c. Double-click *Text in from Control Engine*.



8. Assign the tag.

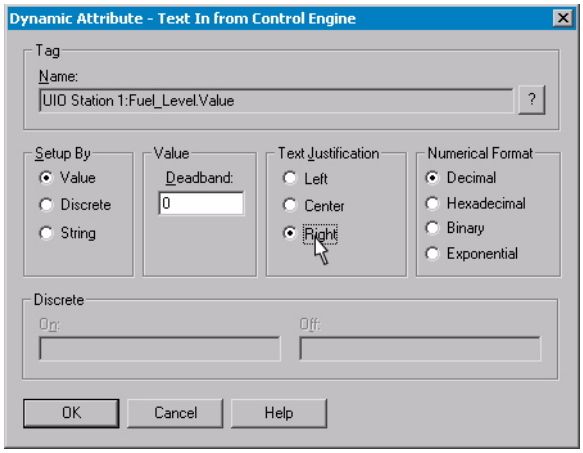
In the Value Tag Selection dialog box, do the following:

a. Make sure *Value* is selected under Setup By.
b. Click the *Tag Selection* button.
c. In the *Item Type* field, select *Analog Input Point*.
d. In the *Item Name* field, select *Fuel_Level*.
e. Click *OK* to close the Value Tag Selection dialog box.



In the Dynamic Attribute dialog box, do the following:

f. For *Text Justification*, choose *Right*.
g. Click *OK* to close the Dynamic Attribute dialog box.
h. Close the Graphic Dynamic Attributes dialog box.



Running Your Project

1. Save your work and go to ioDisplay Runtime.

- Select *File* ➞ *Save Project and Load Runtime*.

If you are making many modifications to an ioDisplay project, it is advisable to save the project at least every 15 to 30 minutes.

2. Test the project.

a. Click *Close* to close the Event Log Viewer.
b. Rotate the tank level counter-clockwise.

   Notice that the bar graph gets smaller and changes color. The numerical value displayed on the bitmap graphic also changes.

c. Turn on the Photo Sensor switch.
d. Rotate the potentiometer clockwise.

   The bar graph gets larger.

e. Turn on and off the Photo Sensor switch and observe the outside light graphic change color.

3. Exit ioDisplay Runtime.

a. Select *Exit ioDisplay Runtime* from the *File* menu.
b. When asked if you want to exit Runtime, select *Yes*.

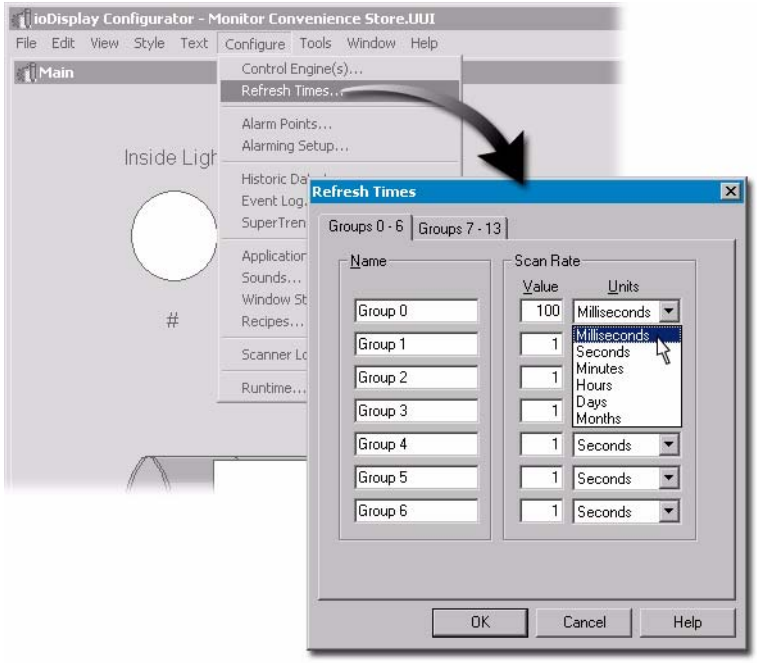
Setting Refresh Times

When you rotated the tank level potentiometer, you may have observed that the display in your HMI was not updated nearly as fast as the meter on your load panel. In this step, you will improve your HMI's performance by changing the properties of a scan group.

A scan group defines how often control engine variables are scanned to refresh ioDisplay graphics. Every ioDisplay tag is assigned a scan group when the tag is selected. There are 14 scan groups with configurable refresh times. Refresh times range from milliseconds to months. Communication loads can be optimized by carefully defining and assigning refresh times.

1. Change the refresh time for Group 0.

a. Select *Refresh Times* from the *Configure* menu.



b. For Group 0, type `100` in the *Scan Rate – Value* field.
c. In the *Scan Rate – Units* field, select *Milliseconds*.
d. Click *OK* to close the Refresh Times dialog box.

2. Save the project and load ioDisplay Runtime.

a. Select *Save Project and Load Runtime* from the *File* menu.
b. Click *Close* to close the *Event Log Viewer*.
c. Test the project by rotating the tank level knobs.

   Notice that the bar graph responds much faster. This is because you changed the rate at which the tag values are updated.

d. When you are finished, choose *Exit ioDisplay Runtime* from the *File* menu.

TRENDING

SKILLS

SuperTrends

- Drawing a SuperTrend object
- Defining trend attributes and scaling
- Configuring trend pens
- Using historical and real-time modes
- Opening historic logs

Operator Interface

- Creating a new window
- Opening and closing windows
- Configuring Runtime options
- Password protecting a window
- Assigning operator-driven attributes

SCENARIO

You now have an operator interface for a computer connected to your SNAP Ultimate I/O. Someone at this computer can monitor devices in the convenience store and control certain points, such as the indoor lighting. This operator interface is a window to the process being controlled by the SNAP Ultimate I/O, and the capabilities of this interface are determined by the developer using ioDisplay Configurator.

In this lesson, you will enhance the monitoring capabilities of your operator interface by using ioDisplay's trending features. Trends allow you to monitor and record changes in values over time. Specifically, your operator interface will be tracking Fuel_Level and Store_Temperature.

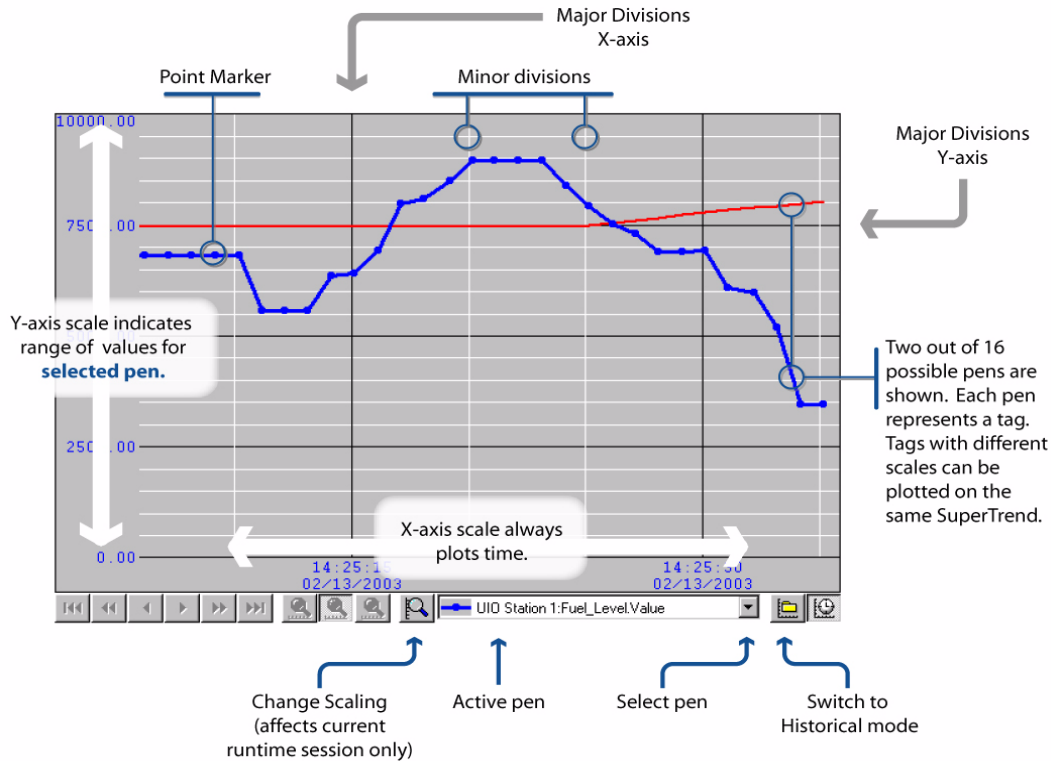
CONCEPTS

SuperTrends

The ioDisplay Tools palette provides Trend, X-Y plot, and SuperTrend tools. Of these, the SuperTrend is the most sophisticated. Using the SuperTrend tool, you draw SuperTrend graphics just as you would any other graphic; however, once created, a SuperTrend allows you to define numerous attributes of a dual-axis graph, where the X-axis is

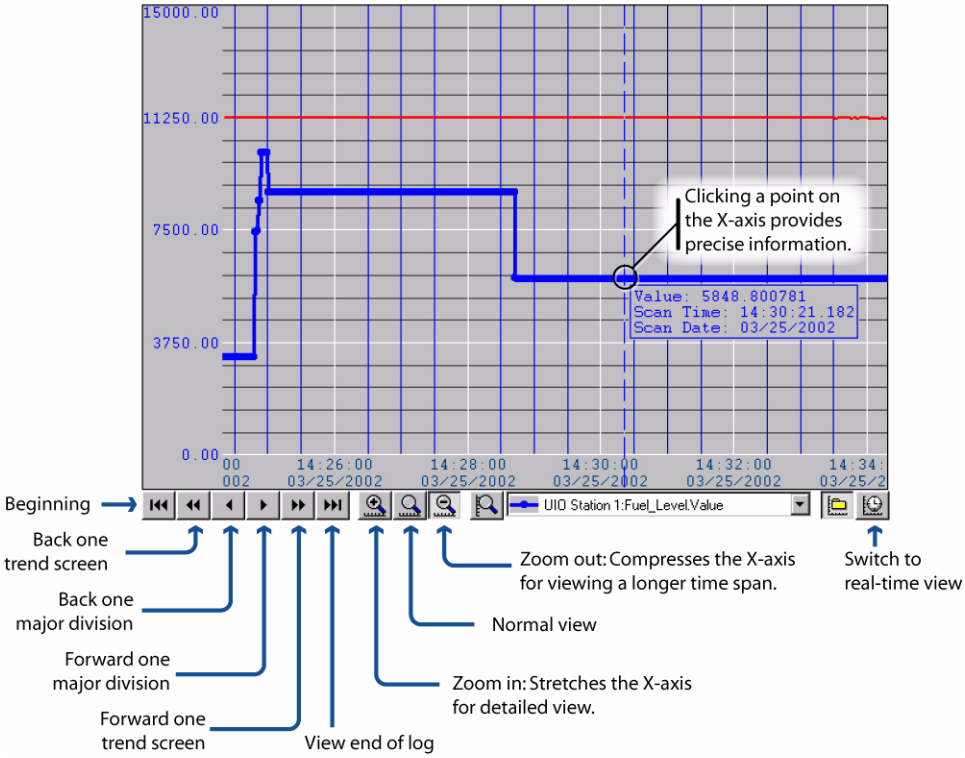
always time and the Y-axis is a scale you define. SuperTrends can contain up to 16 tags and display your tags in real-time and historical modes.

SuperTrends: Features of Real-time Mode



Historical mode creates log files that can be opened at any time for analysis.

SuperTrends: Features of Historical Mode

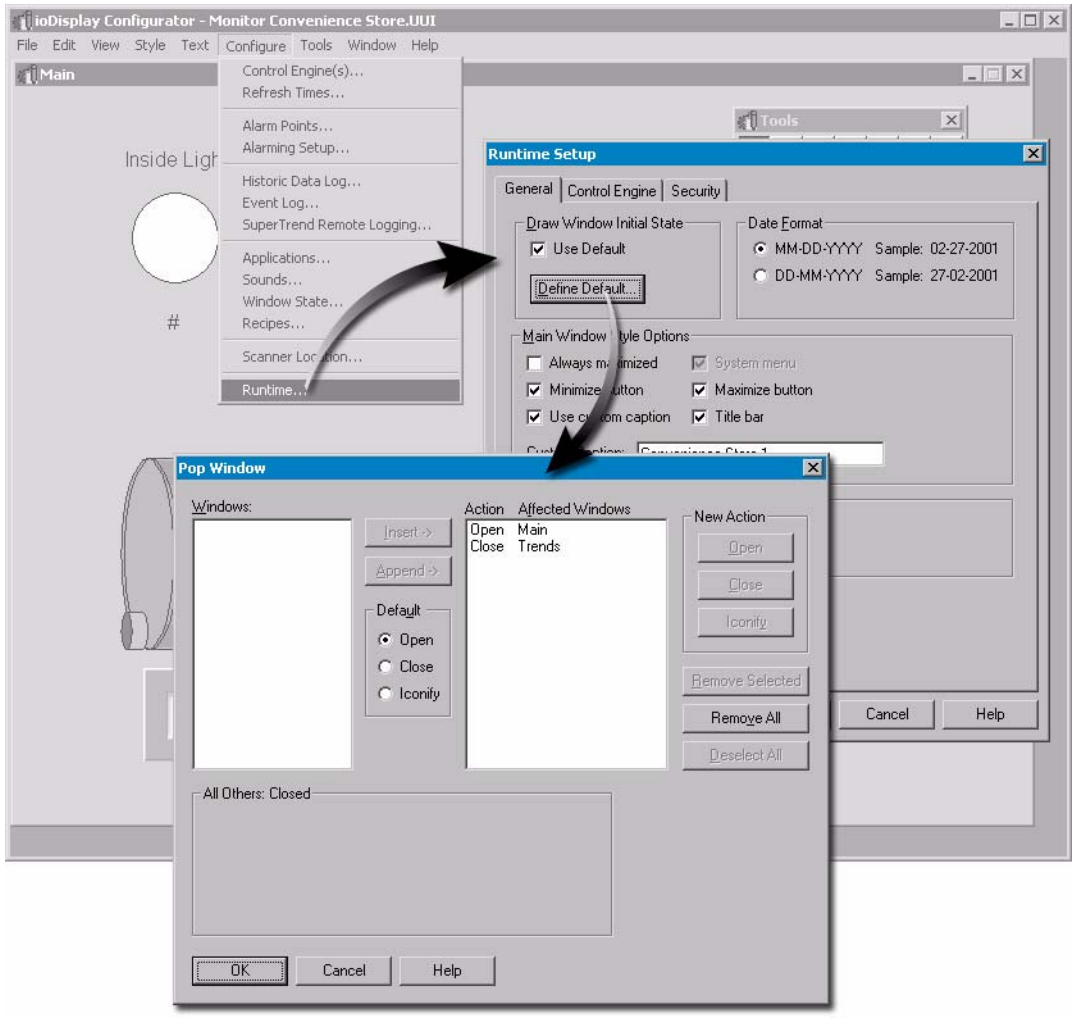


About ioDisplay Windows

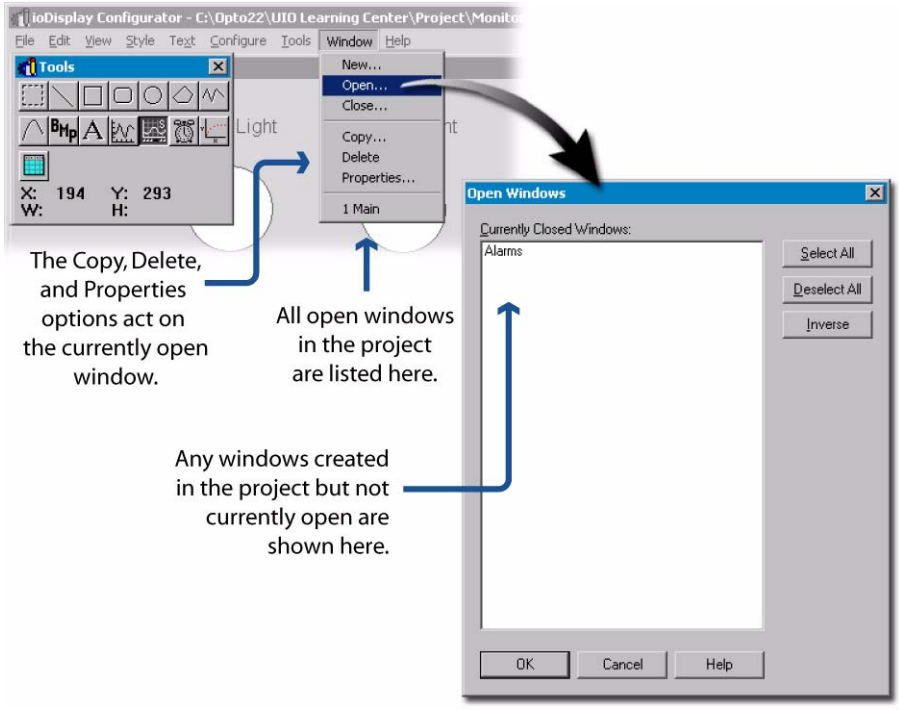
In this and the following lesson, you will add additional windows to your project. As you work in ioDisplay Configurator, the appearance of your project closely mimics the runtime result. It is easy to forget that some features in the ioDisplay Configurator user interface are actually controlling the ioDisplay Runtime interface. For example, if you open a window to contain the SuperTrend you will build in this lesson, you should make a decision as to whether you want this window open all the time or just upon certain actions. If you leave it open in Configurator, it will be open by default in Runtime. It may be inconvenient to have to open or close your windows to ensure the

desired defaults in runtime, so ioControl gives you a way of configuring the default state of each window. The location of the configuration settings is shown below:

Features used to set the default states of windows:



Windows can be opened or closed as needed by the Window menu and as an operator driven attribute. Here are some techniques to remember when working with multiple windows in ioDisplay Configurator:



ACTIVITY

Preparing for the Lesson

Make sure your ioControl strategy is loaded and running on the SNAP Ultimate controller/brain.

You can use the ioProject utility ioTerminal to check if your strategy is running. (See 'Preparing for the Lesson" on page 2-5).

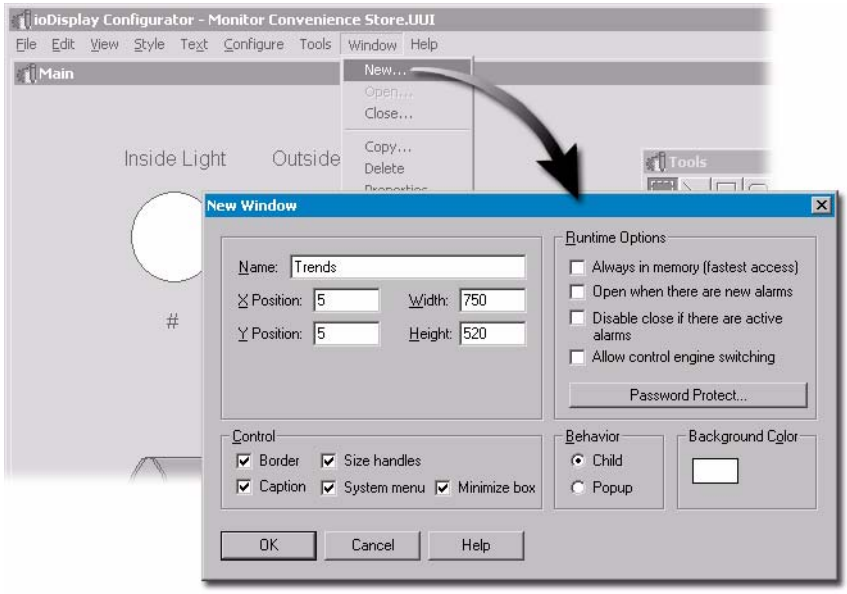
Make sure ioDisplay is running and your project is open.

Creating a New Window

1. Create a new window.

Select *Window* ⟶ *New*.

2. Configure the window's attributes.



a. Provide the following information in the New Window dialog box:

Name field: `Trends`

X Position: `5`

Y Position: `5`

Width: `750`

Height: `520`

b. Check the *System menu* option box.

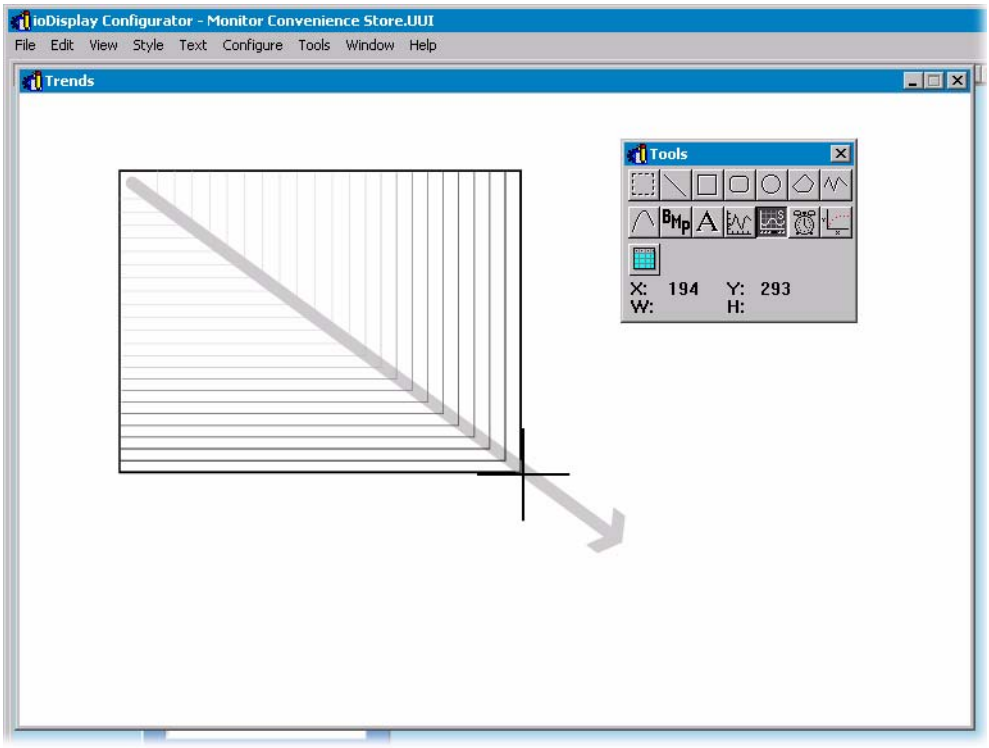
This option places the control icon in the upper left of the window and the minimize, maximize, and close icons in the upper right of the window.

c. Click *OK* to close the New Window dialog box.

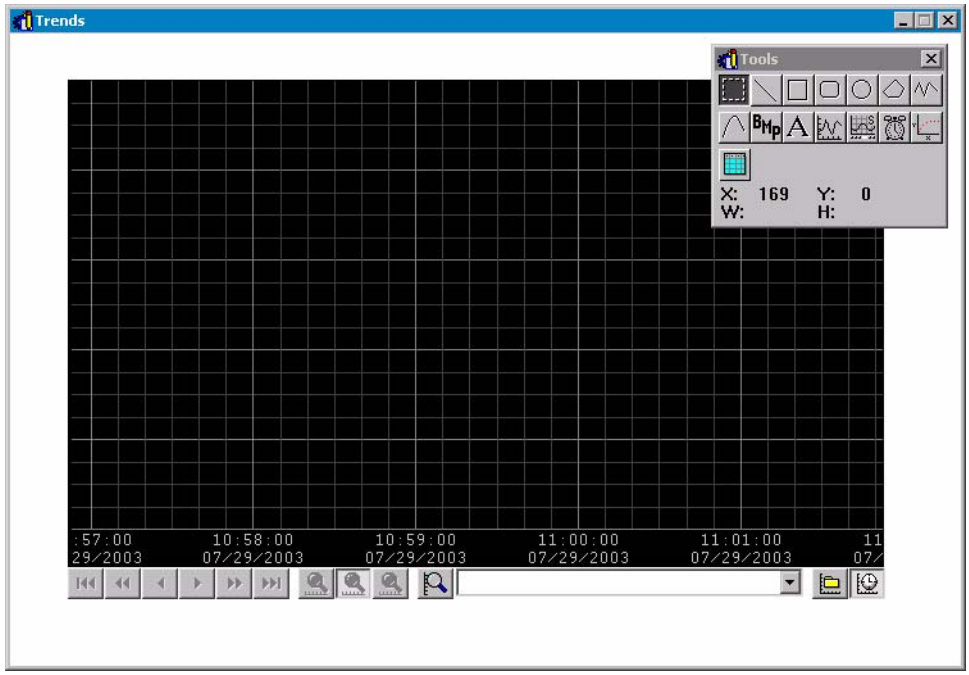
Creating a SuperTrend

1. Draw a SuperTrend.

a. Select the *SuperTrend* tool from the Tools palette.
b. Click and drag in the Trends window to draw the SuperTrend.



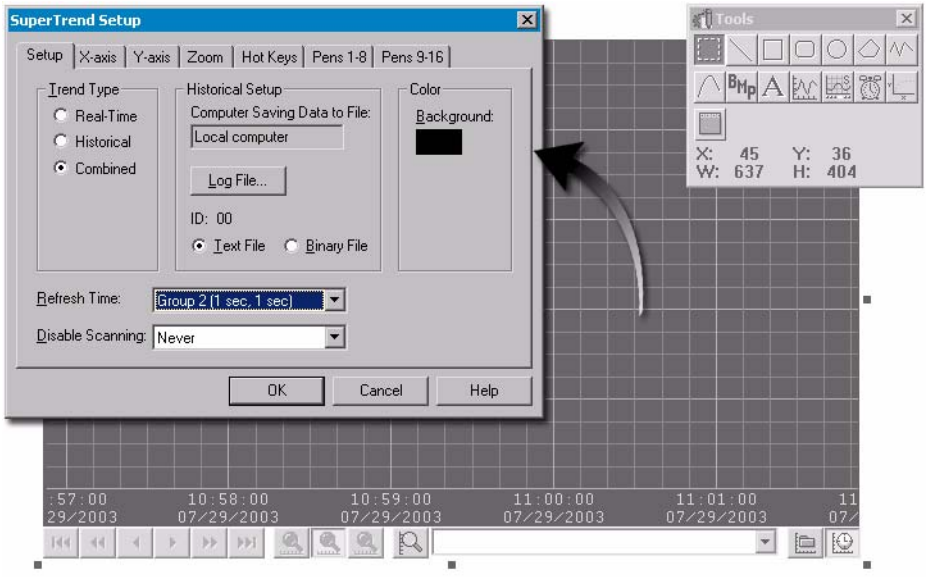
Your SuperTrends graphic should appear as shown here:



Use the Select tool if you need to resize your SuperTrend. In order to match the screen shown here, your SuperTrend needs to be wide enough to display all of its icons on one line.

2. Select SuperTrend setup options.

a. Choose the *Select* tool.
b. Double-click the *SuperTrend*.



c. In the *Trend Type* field, make sure that *Combined* is selected.
This SuperTrend will plot both real-time and historical data.

d. In the *Refresh Time* field, select *Group 2*.
e. In the *Disable Scanning* field, select *Never*.

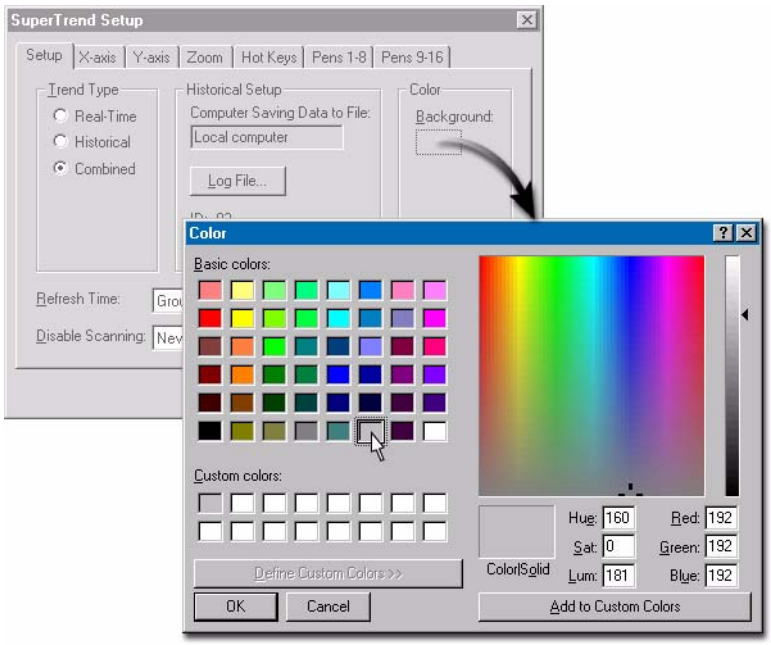
> **About Scanning**
>
> A trend continuously polls for data unless it is disabled. There are options to disable the trend when the window is closed and/or iconified. These options improve efficiency when numerous trends are used. If every trend is continually scanning, the system throughput is slowed. Disabling the scanning reduces the communications overhead between ioDisplay and the control engine.
>
> When a trend is disabled, it does not plot data. When the trend is re-enabled, scanning begins automatically, but there will be a gap in the trend data.
>
> In this example, we are using one SuperTrend that monitors gradual change. So there is no need to optimize performance.

3. Change SuperTrend background color.

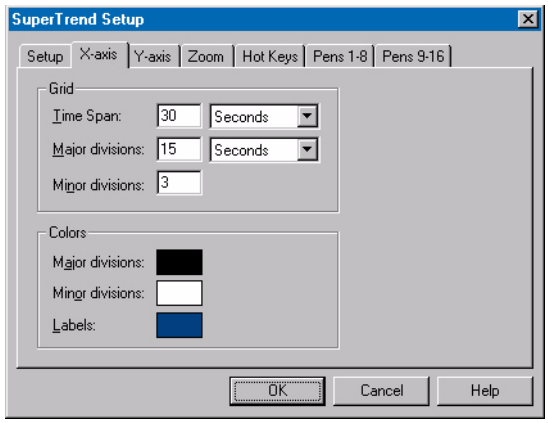
a. Click the *Color – Background* box.

This opens the Color dialog box.

b. Choose a light gray color.
c. Click *OK* to close the Color dialog box.



4. Select X-axis options.



a. Click the *X-axis* tab.

b. In the Time Span field, enter `30 Seconds`.

   The trend will display 30 seconds of data at one time.

c. In the Major divisions field, enter `15 Seconds`.

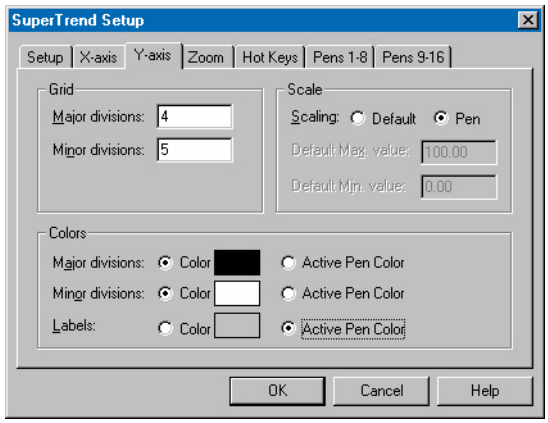
   The trend will have a major division every 15 seconds.

d. In the Minor divisions field, type 3.

   Each major division will have 3 minor divisions. For this example, there will be a minor division every 5 seconds.

e. In the Colors section, change the Major divisions color to black, the Minor divisions color to white, and the Labels color to dark blue. These colors provide good contrast to the light gray background of the trend.

5. Select Y-axis options.



a. Select the *Y-axis* tab.

b. In the Grid section, type `4` in the Major divisions field and `5` in the Minor divisions field.

c. Under Scale make sure the Scaling is set to *Pen*.

d. In the Colors section, change the Major divisions color to black and change the Minor divisions color to white.

e. For the Labels color, select the *Active Pen Color* option.

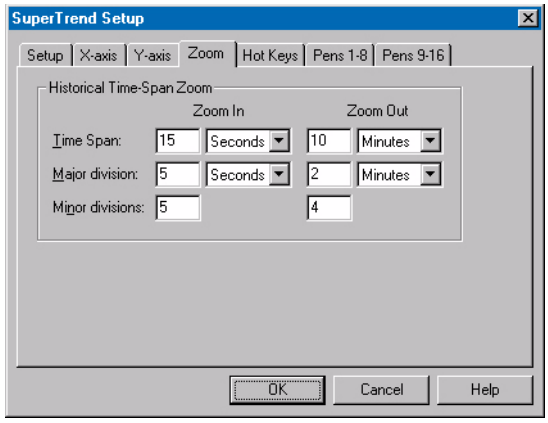
   The labels color will be the same as the color used to plot a selected tag.

6. Select zoom options.

The Zoom options allow you to configure the zoom levels for the X-axis in historical mode.

a. Select the Zoom tab.

The Zoom In level lets you look at a more detailed section of a historic log. The Zoom Out level lets you look at a less detailed section of a historic log.



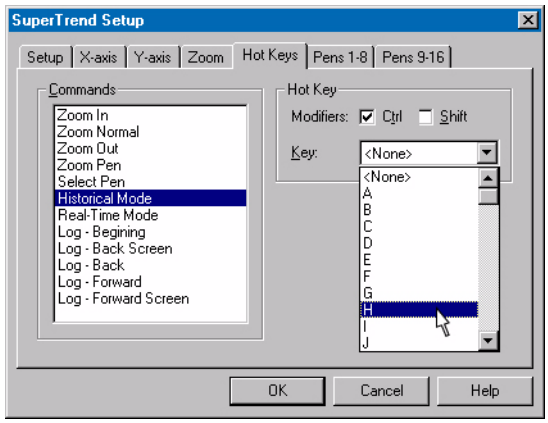
b. In the Zoom In section, enter `15 Seconds` as the Time Span.

The Time Span is the amount of time displayed in the grid when in Zoom In mode.

c. Enter `5 Seconds` as the Major division and `5 seconds` as the Minor division.

d. In the Zoom Out section, enter `10 Minutes` as the Time Span, `2 Minutes` as the Major division, and `4` as the Minor division.

7. View hot keys configuration options.



a. Select the *Hot Keys* tab.

You can assign a hot key to one of the commands listed. In Runtime, typing an assigned hot key performs the same action as clicking a SuperTrends command button.

b. Under Commands, select *Historical Mode*.

c. Under Hot Key select *Ctrl*.

d. In the Key dropdown list, select *H*.

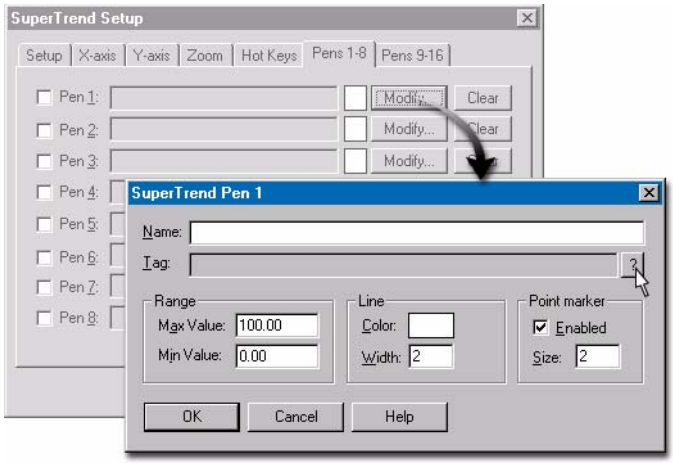
e. Click OK.

8. Configure Pens.

a. Select the *Pens 1–8* tab.

This dialog shows all of the pens, whether or not they are enabled, and the color of the pen.

b. Click the *Modify* button for Pen 1.

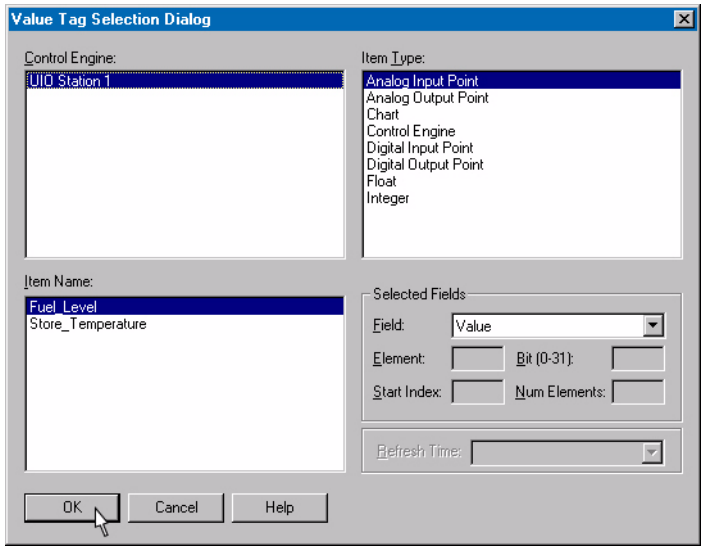


c. Click the Tag Selection button.

d. Select *Analog Input Point* in the Item Type field.

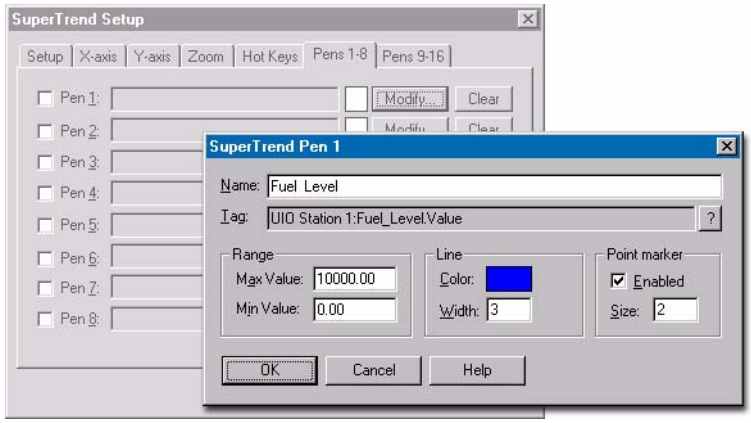
e. Select *Fuel_Level* in the Item Name field.

f. Click *OK* to close the Value Tag Selection dialog box.



g. Type `Fuel Level` for the Name.

(Note that you can use spaces in pen names.)



h. Type `10000` for the *Max Value*. This value describes the value of the line when it is at the top of the trend.
i. Type `0` for the *Min Value*.
j. Select a bright blue color for the *Line Color*.
k. Type 3 for the *Line Width*. This value describes the width of the line in pixels.

   Make sure the Point marker is enabled. A marker will be displayed for every scanned point. This feature makes it easy to identify scanned data.

l. Click *OK*.

9. Configure Pen 2.

a. Configure Pen 2 just as you did with Pen 1, but enter the following information:

   Name: `Store Temperature`

   Tag (Open the Value Tag Selection dialog box):

   Item Type: *Analog Input*

   Item Name: *Store_Temperature*

   Field: *Value*

   Pen:

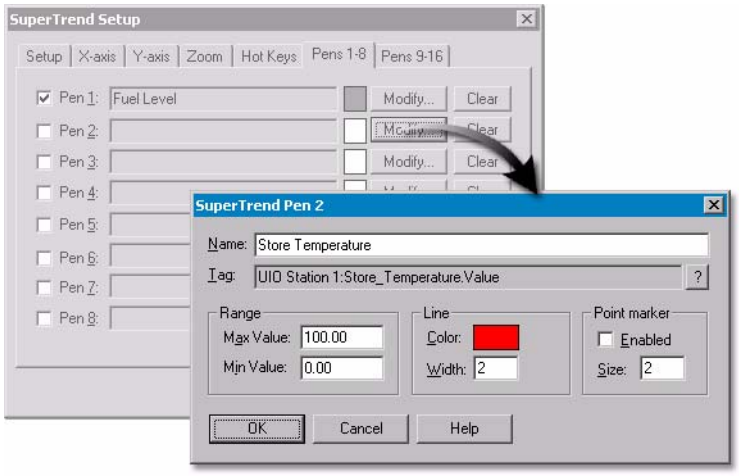
   Max Value: `100`

   Min Value: `0`

   Line Color: Red

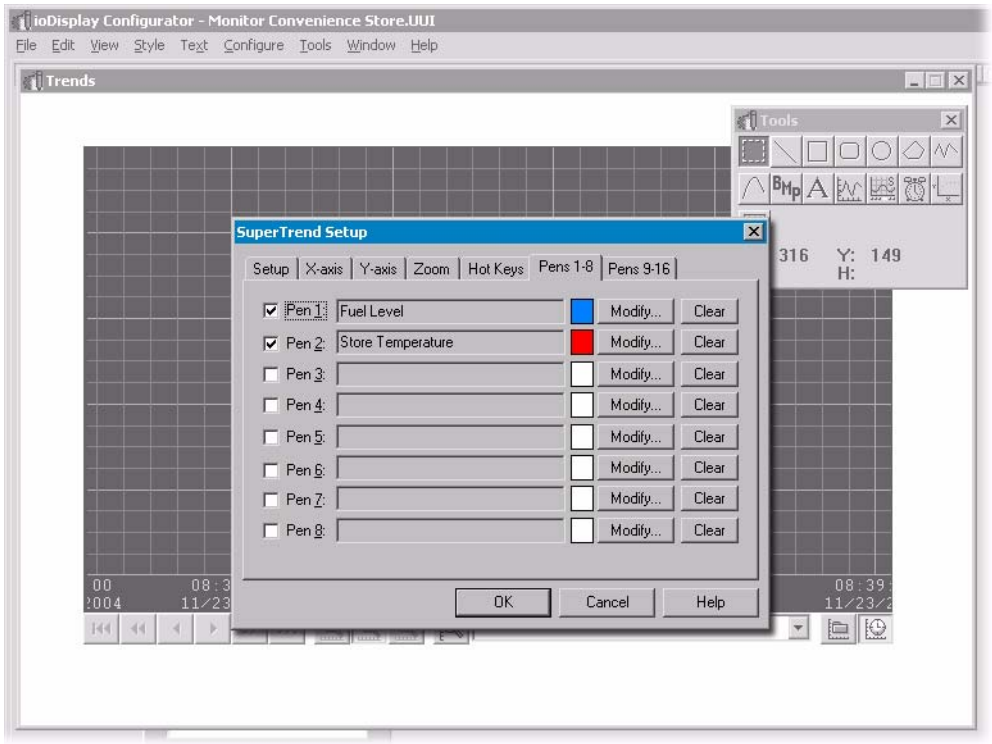
   Line Width: 2

b. Disable the *Point marker*.



c. Click *OK* to close the SuperTrend Pen 2 dialog box.

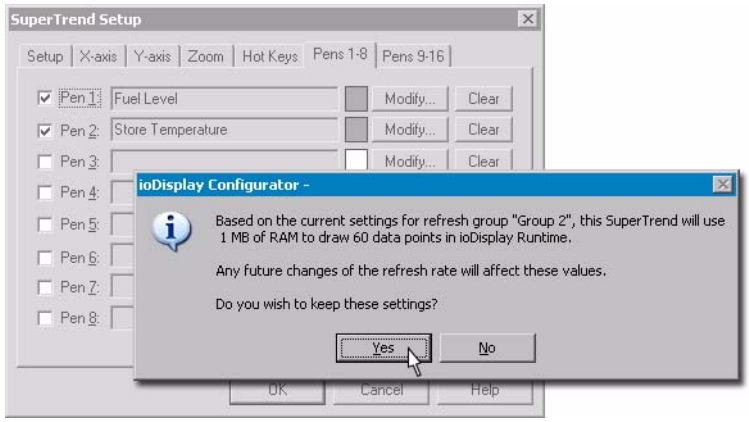
10. Verify that you have two pens configured and enabled.



11. Close the SuperTrend Setup dialog box.

a. Click *OK*.

A message on memory usage by the new SuperTrend is shown.



b. Click *Yes*.

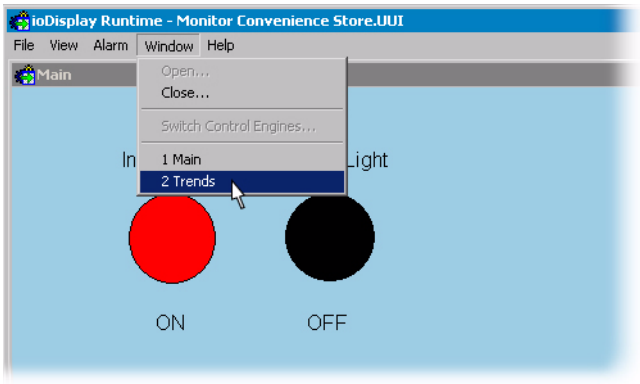
Testing Your Changes

1. Save your project and load ioDisplay Runtime.

a. Select *File* ➞ *Save Project and Load Runtime*.
b. Close the Event Log Viewer.

2. Open the Trends window.

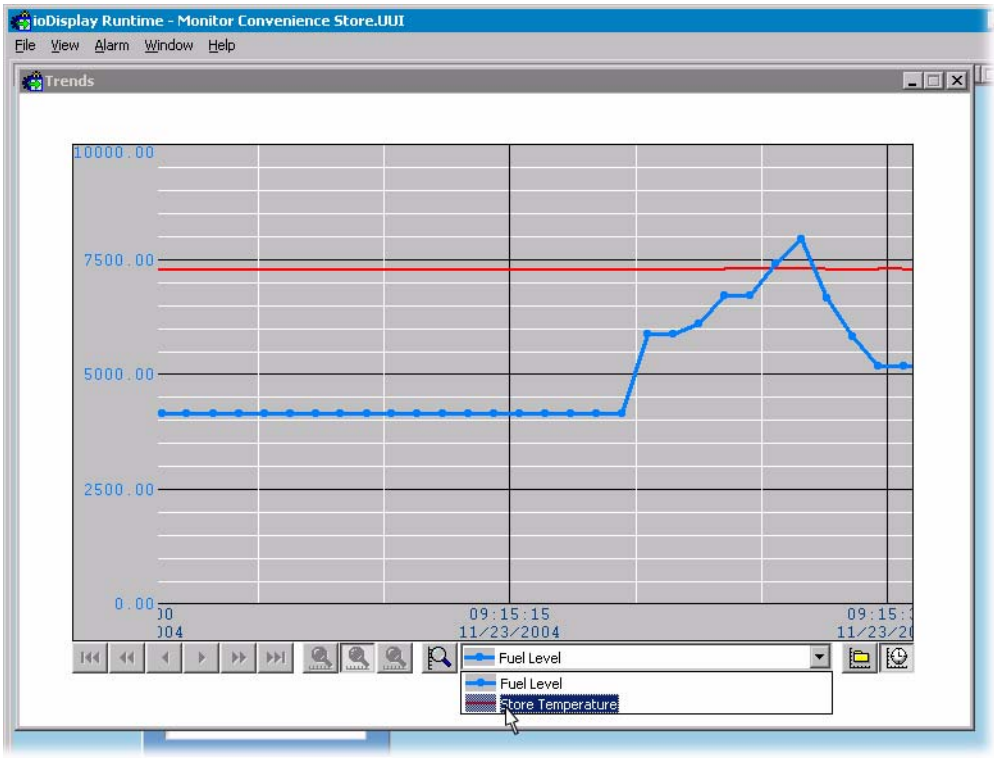
a. Select *Window* ⟶ *Trends*.



NOTE: Most of the buttons at the bottom of the trend are disabled. You will use these buttons when you switch to the Historic mode later in this exercise.

The scale on the left-hand side of the graph is based on the active pen. The active pen is listed in the drop-down box at the bottom of the graph.

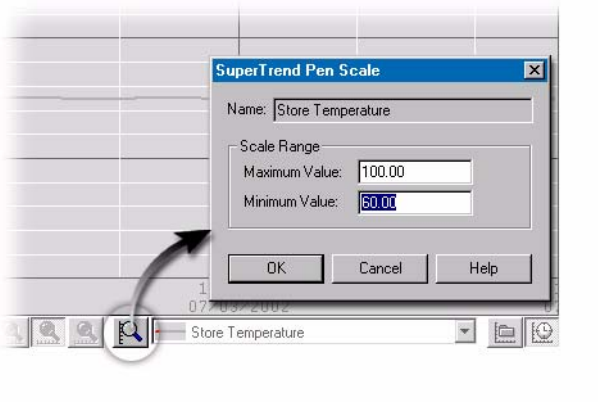
b. Rotate the Fuel Level potentiometer and notice the change to the SuperTrend.



c. Click the drop-down menu at the bottom of the SuperTrend.

d. Select *Store Temperature*.

Notice that the scale on the Y-axis changed to match the scaling of the Store_Temperature tag. The scaling was set in ioDisplay Configurator, but you can change the scaling in Runtime by clicking on the Scale button.



e. Click the *Scale* button.

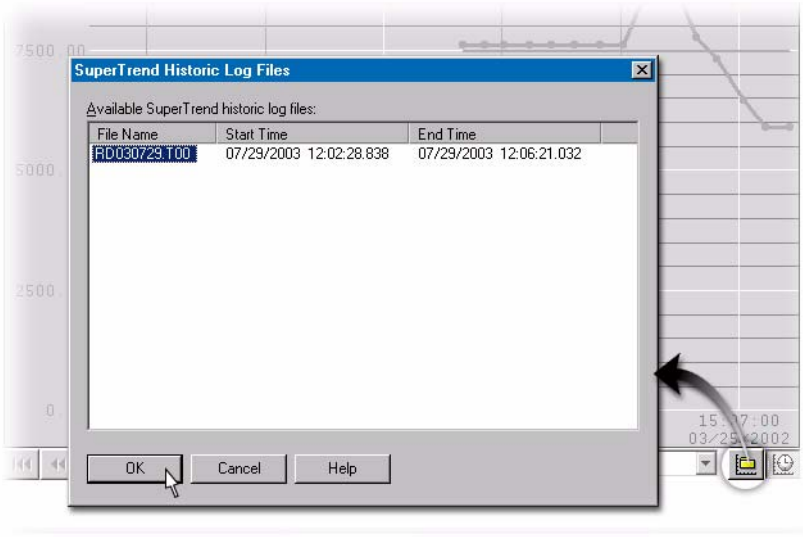
f. Type `60` in the Minimum Value field.

g. Click *OK* to close the SuperTrend Pen Scale dialog box.

Notice that the SuperTrend X-axis is showing a scale of 60 to 100.

3. Examine the historic trend.

a. To switch to Historic mode, click the *Historic Mode* button ( ).

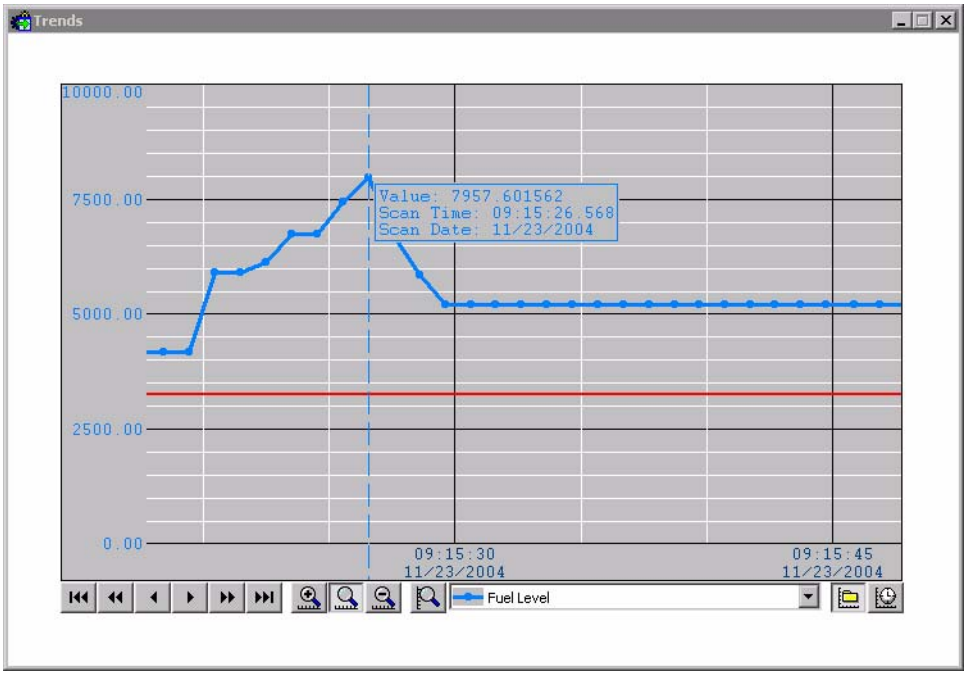


The SuperTrend Historic Log Files dialog box lists all available SuperTrend files. Start and end times are listed for each file, so you can choose the desired file. Because the specified SuperTrend file rollover time is longer than the time ioDisplay has been trending, you may have only one file.

b. Click the file to select it, and click *OK*.

c. Experiment with your trend. Scroll forward and backward. Zoom in and out. Change the scale by selecting a new active pen.

d. Click on a point marker on the trend plotted by the active pen. ioDisplay pops up a box that lists the point's value and time/date stamp.



e. Switch back to the real-time trend by clicking the *Real-time Mode* button.

Notice that the real-time plot has no gaps in the data. ioDisplay continues to record and plot data, even while in historic mode.

NOTE: If your SuperTrend is not displaying a tag, you may have not scaled the pen correctly. For example, if a tag's value is 7500, but the scale is the default 0-100, the pen would be off the scale.

4. Exit Runtime.

a. When you are finished, select *File* ⟶ *Exit ioDisplay Runtime*.
b. Click *Yes*.

Using Windows in the Operator Interface

The SuperTrend is in its own window, and you used ioDisplay Runtime's window menu to view the Trends window.

Choosing *Window ⟶ Trends* brought the Trends window to the front. If you closed your Trends window in Configurator, closed would have been the default state, and you would have used *Window ⟶ Open* to open your Trends window. ioDisplay used the last state of the windows as the default.

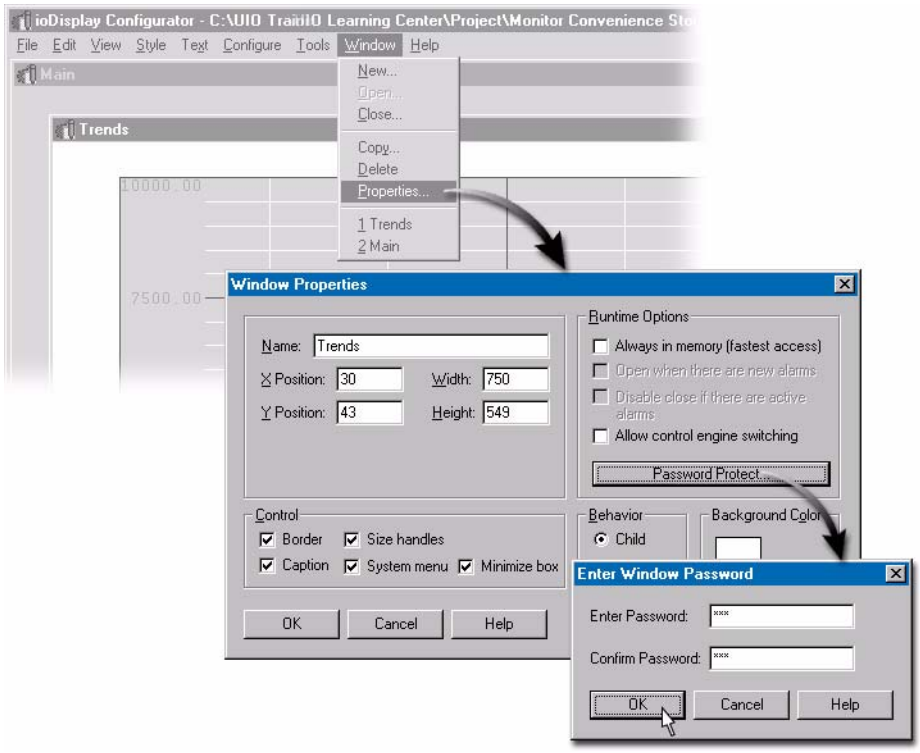
Next, you will:

- Configure the initial state of the window.
- Add a button within the Main window that opens the Trends window.
- Add Password protection for the Trends window.

1. Password protect the Trends window.

a. Bring the Trends window to the front, if it is not there already.
b. Choose menu *Window ⟶ Properties*.
c. Click *Password Protect*.
d. Type a password in both fields.

NOTE: Passwords are case-sensitive.

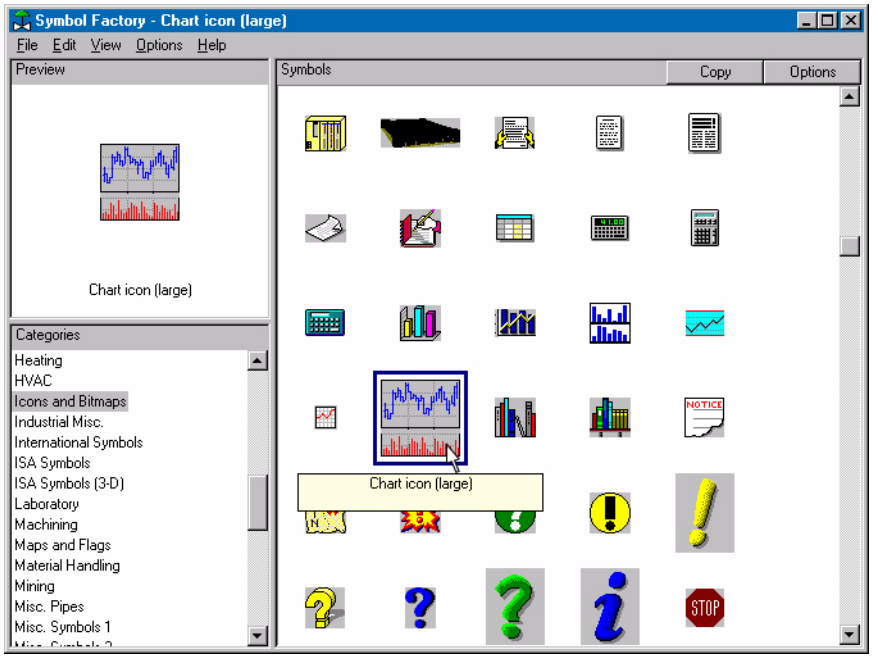


e. Click *OK*.
f. Close the Windows Properties dialog box.

2. Copy a button graphic from Symbol Factory.

a. In ioDisplay, choose *Edit ⟶ Paste from File ⟶ Symbol Factory*.
b. Choose the category *Icons and Bitmaps*.

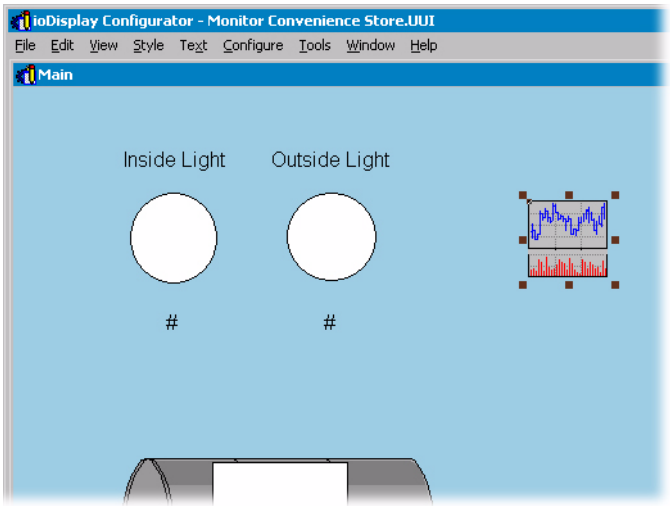
c. Select the Chart icon and click *Copy*.



d. Close Symbol Factory.

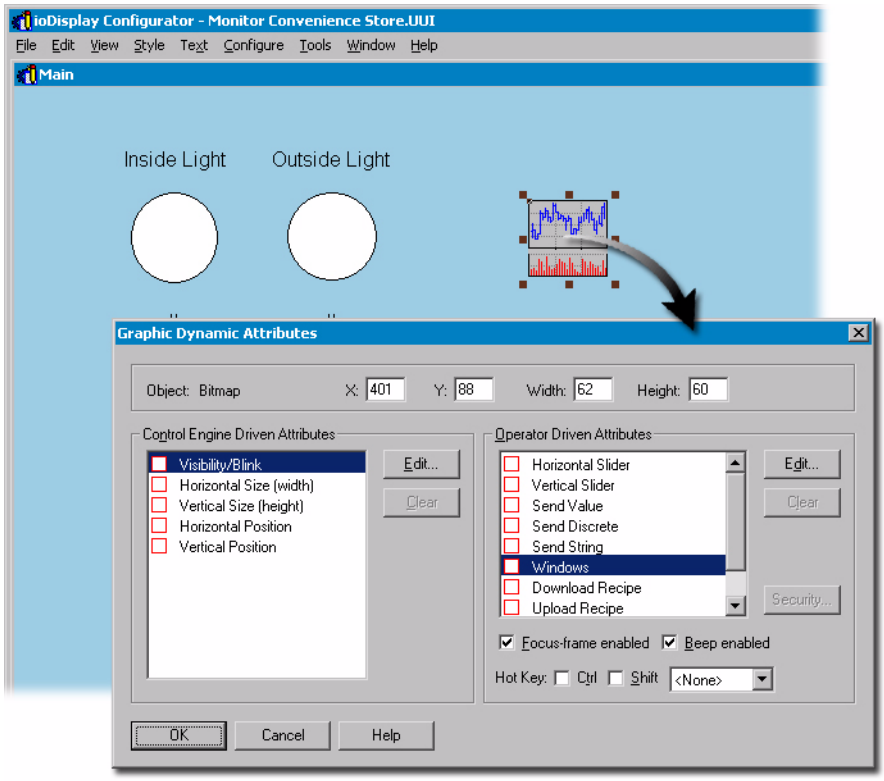
3. Paste the icon into the Main window.

a. Bring the Main window to the front.
b. Choose *Edit* ⟶ *Paste*.
c. Drag the graphic to the desired location.



4. Assign operator-driven attributes to the icon.

a. Double-click the chart icon.



b. Double-click *Windows* to open the Pop Window dialog box.

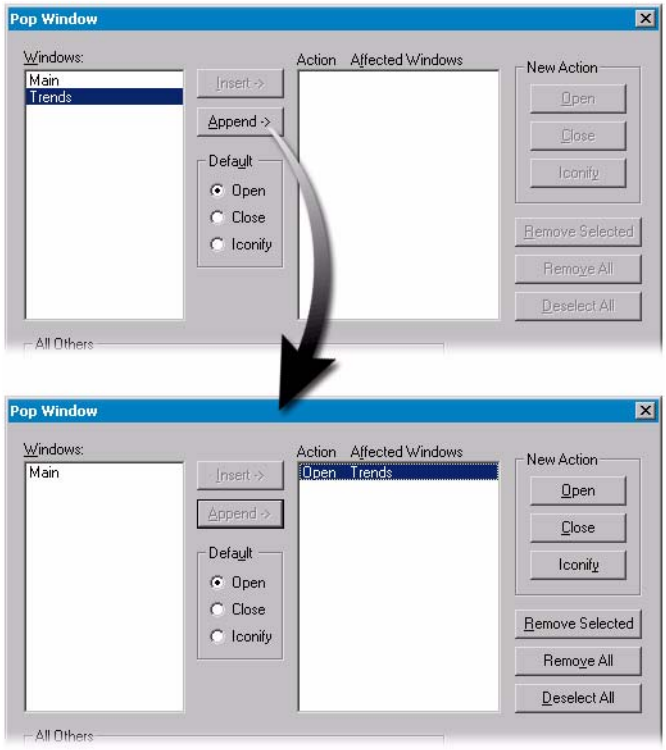
c. In the Windows field, select *Trends*.

The default action *Open* is already selected.

d. Click *Append*.

The action is now described under the Action Affected Windows field.

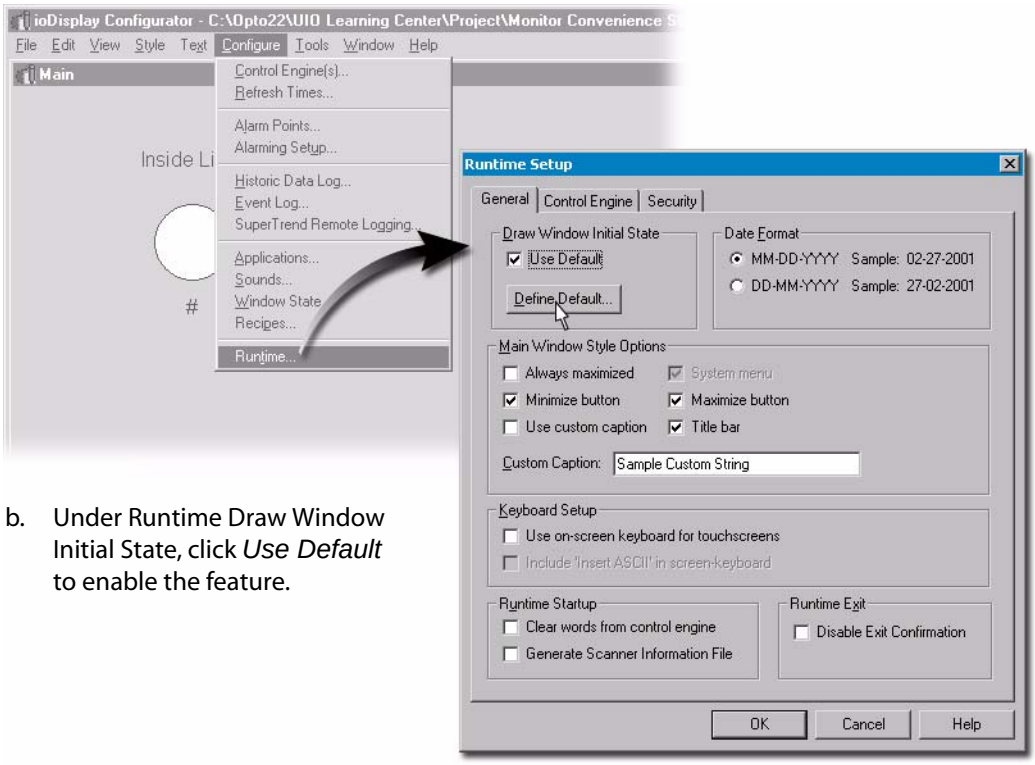
e. Close all dialog boxes.



5. Open the Runtime Setup Options dialog box.

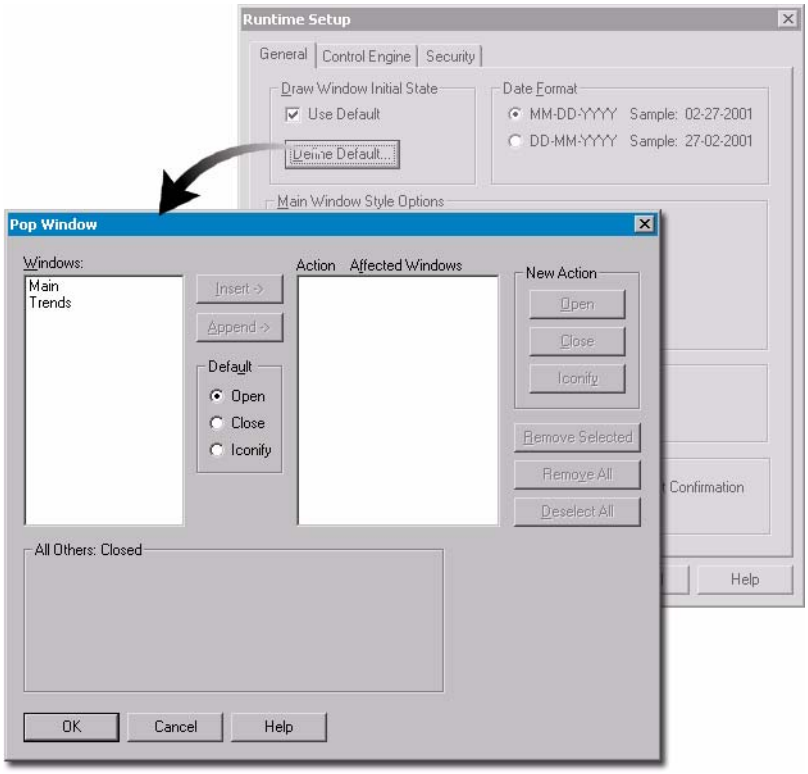
We want the Trends window to be closed at startup.

a. Select *Configure* ➞ *Runtime*.



b. Under Runtime Draw Window Initial State, click *Use Default* to enable the feature.

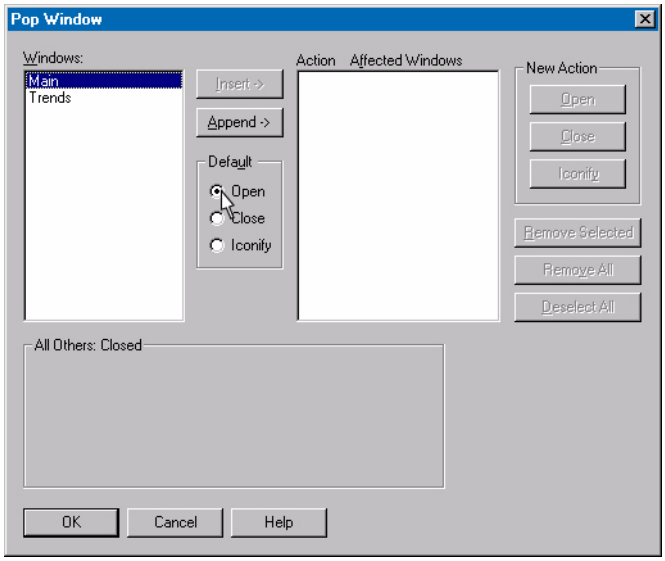
c. Click the *Define Default* button.



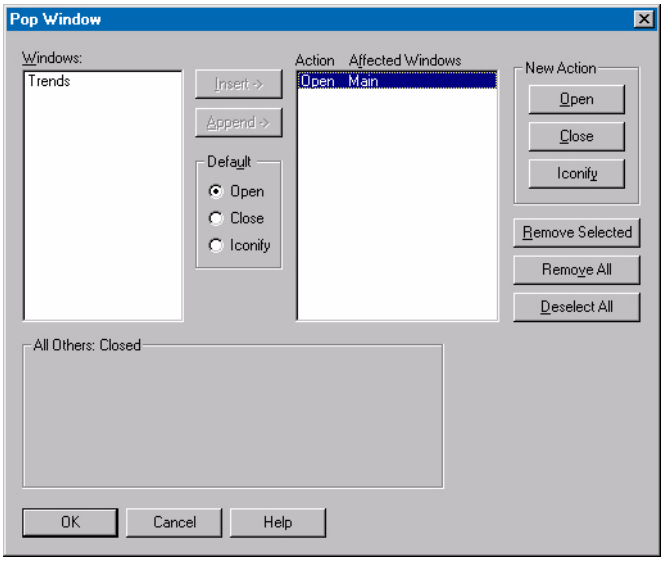
The windows you defined are listed in the *Windows* field. To set a default action for a window, you select the window, chose *Open*, *Close*, or *Iconify*, and then click *Append*.

6. Define the default state of the Main window.

a. In the Pop Window dialog box, select *Main*.
b. In the *Default* group, select *Open*.



c. Click *Append*.

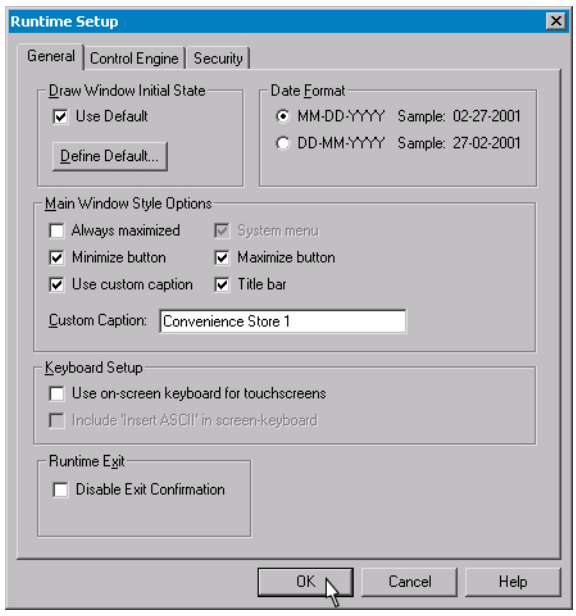


7. Define Trends window caption.

a. Under *Main Window Style Options*, select *User custom caption*.

b. In the Custom Caption field, type `Convenience Store 1`.

c. Click *OK* to close the Runtime Setup dialog box.



Testing Your Changes

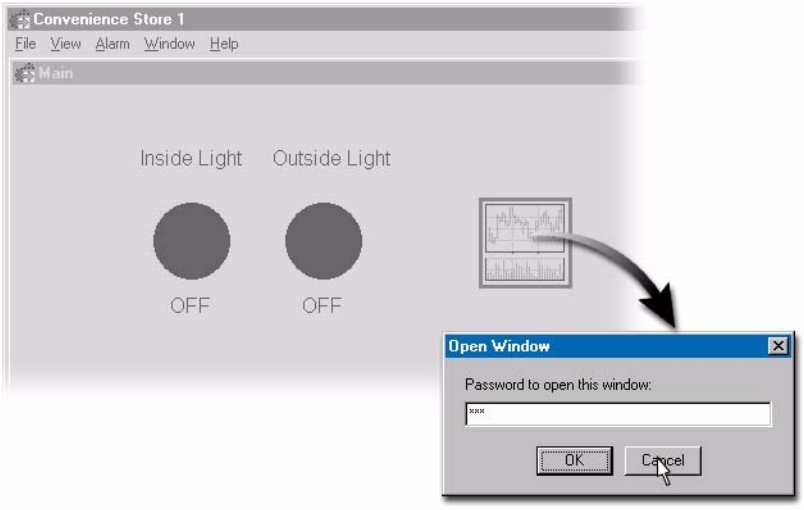
1. Save and run your project.

Choose *File* ➞ *Save Project and Load Runtime*.

2. Open your Trends window.

a. Click your Chart icon.

b. Type your password and click *OK*.



3. View your trend.

4. Close ioDisplay Runtime after viewing your SuperTrend.

ALARMING

SKILLS

Alarms

- Configuring alarm points
- Creating alarm graphics
- Using alarm types: Detailed, Summary, History

Operator Interface

- Enabling alarms within a drawing window
- Opening and closing drawing windows

SCENARIO

In developing your store monitoring system, you may wish to alert the operator to a situation that needs action. For example, should the fuel level be low, you would want the operator to arrange a fuel delivery. In this lesson, you will create alarms that alert an operator to the following conditions that may need human intervention:

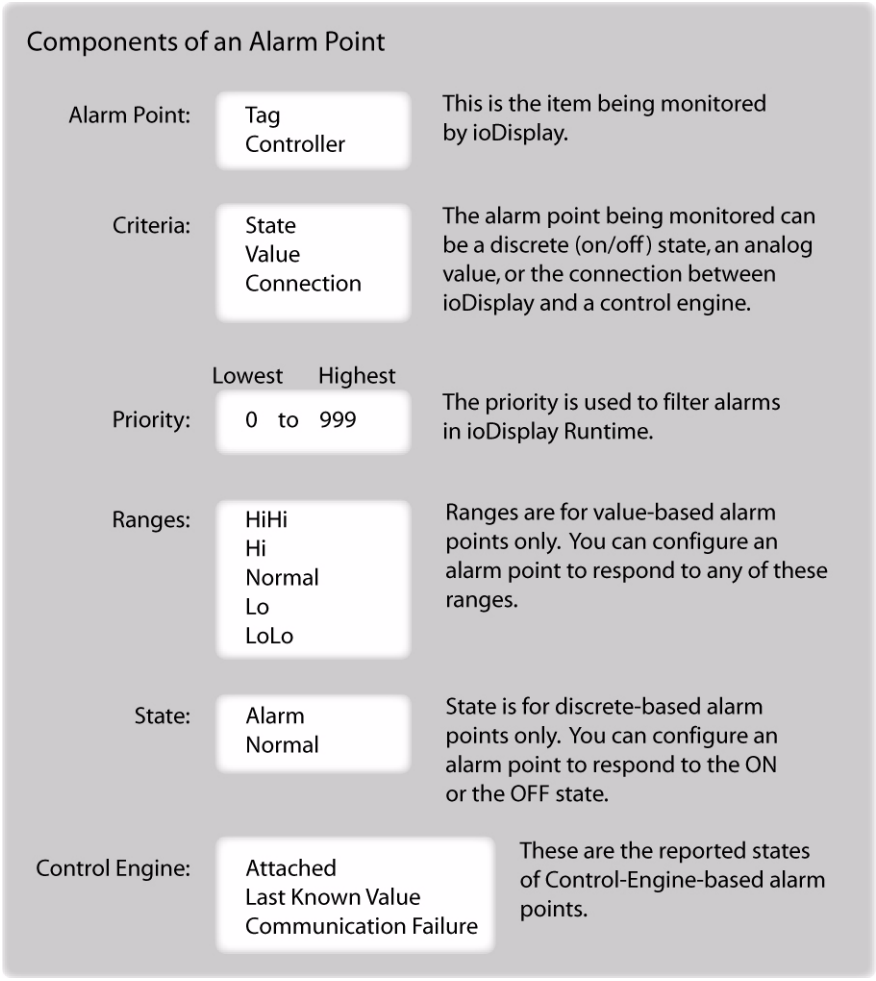
- Low fuel level
- Freezer door left open
- Lost connection to the control engine

CONCEPTS

Alarm Points

Alarm points are the basis of the alarming feature. Each alarm point is linked to an ioControl tag. In ioDisplay Runtime, the tag value of each alarm point is compared to the configured alarm state. When an alarm point is in an alarm state, it is displayed on all alarm graphics that contain that alarm point. It will also be sent to any configured file or printer logs. Alarm points are global to a project and can be included in any number of alarm graphics. It is possible that a configured alarm point will never be included in

an alarm graphic, but its tag will still be scanned and sent to any configured file or printer logs.

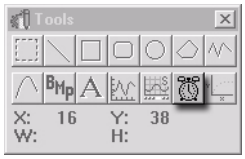


Alarm Graphics

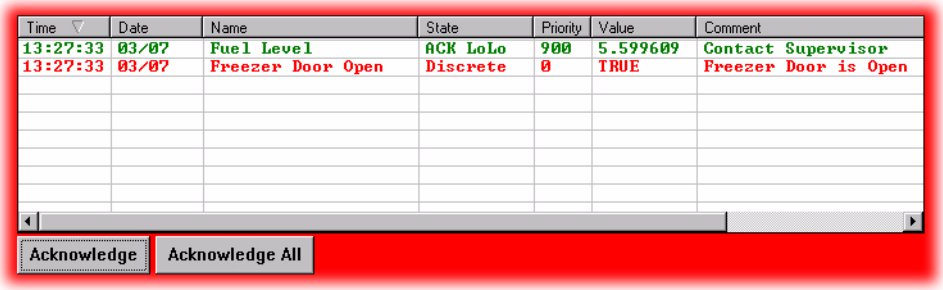
An alarm graphic provides the human interface to the alarm points. You create an alarm graphic using the Alarm Tools on the tool palette and configure it to contain alarm points. More than one alarm graphic can be placed in any ioDisplay window and can be

resized. Alarm graphics use Windows listview controls; for example you can double-click a column heading to sort the list by that column's entries.

Alarm Graphics



Alarm tool is used to draw alarm graphics.



Alarms contain information from the alarm point configuration and include the time, data, and value or state of the alarm point. Column headings also sort the list.

Alarm graphics can display the time and date that the alarm occurred, as well as the name and state of the point, its priority, and a comment you define.

There are three types of alarm graphics:

- **Detailed**—Detailed alarm graphics treat each alarm state as a different alarm condition. For example, consider an alarm that went from its normal state to the Hi state, and then to the HiHi state. The detailed alarm graphic would have separate alarm notices for both the Hi and HiHi alarm states. Both of these states must be acknowledged before they are removed from the alarm graphic.

  Detailed alarms have buttons for acknowledging and silencing alarm points. The Acknowledge button will acknowledge all selected alarm points. The Acknowledge All button will acknowledge all of the alarm points. Silence and Silence All will turn off an alarm until it is unsilenced. Hot keys can be configured to Acknowledge, Acknowledge All, Silence, and Silence All as well.

- **Summary**—Summary alarm graphics are similar to detailed alarm graphics, except that only the most current state for an alarm point is displayed. For example, suppose an alarm point goes from its normal condition to the Hi state. The summary graphic would display the Hi state alarm. If the alarm point then enters the HiHi state, the graphic will be updated and display only the new HiHi state.

  Summary alarms have two buttons to acknowledge alarm points. The Acknowledge button will acknowledge all selected points. The Acknowledge All button will acknowledge all of the alarm points. Hot keys can be configured to Acknowledge or Acknowledge All as well.

- **History**—The History alarm graphic displays a log of all its alarm point activity. There are no acknowledgement buttons on History alarm graphics.

File Logging

Alarm points can be logged to a text file. All alarm points are logged. A subset of alarm points cannot be specified.

Printer Logging

Alarm points can be logged to one printer or to multiple printers. These printers must be installed on the computer. Since Windows 2000 and Windows XP use different naming schemes for printers, you will need to reconfigure printers in the project when moving between operating systems. All alarm points are printed. A subset of alarm points cannot be specified.

Sounds

ioDisplay can be configured to play a sound once when an alarm point enters an alarm state, or to play continuously while any alarm points are active. The sound file can be located in the project directory or have a fixed path. To use this capability, your PC must have a properly configured sound card and driver to support the desired sound file types. Both *.wav and *.mid files are supported.

ACTIVITY

Preparing for the Lesson

1. Make sure your ioControl strategy is downloaded and running on your SNAP Ultimate I/O system.

2. Open your ioDisplay project in ioDisplay Configurator.

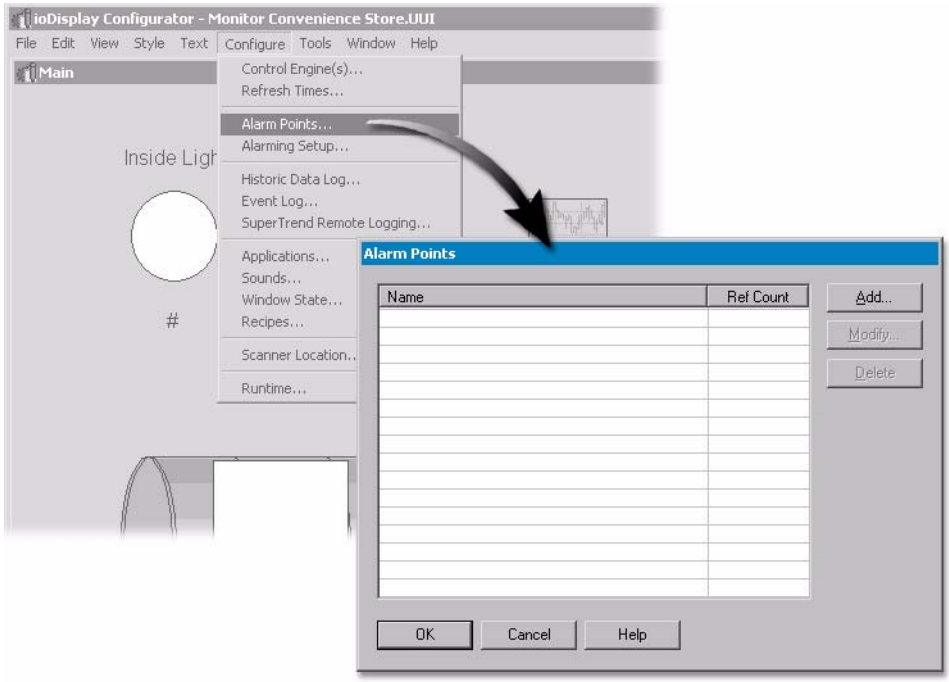
Configuring the Fuel Level Alarm Point

The first part of creating alarms is defining the alarm points. An alarm point's configuration consists of the following:

- A tag that you use to indicate an alarm state, for example, Fuel_Level.
- Criteria for evaluating the tag, such as what value for Fuel_Level should trigger an alarm.
- Priority of the alarm. ioDisplay lets you prioritize alarm points on a scale of 0–999.

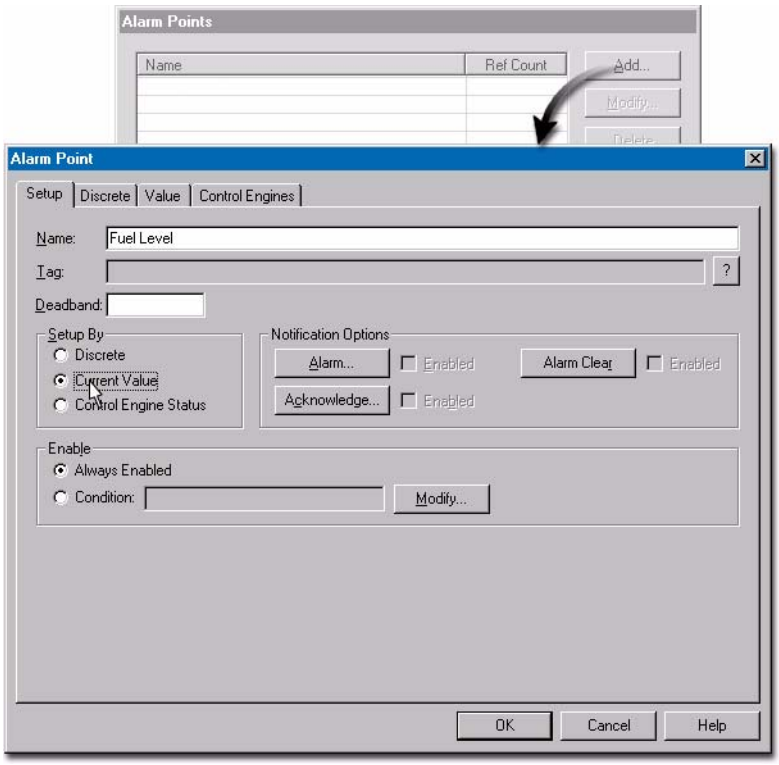
1. Open Alarm Points Configuration dialog box.

Select *Configure* ⟶ *Alarm Points*.



2. Create a new alarm point.

a. Click *Add* to open the Alarm Point dialog box.



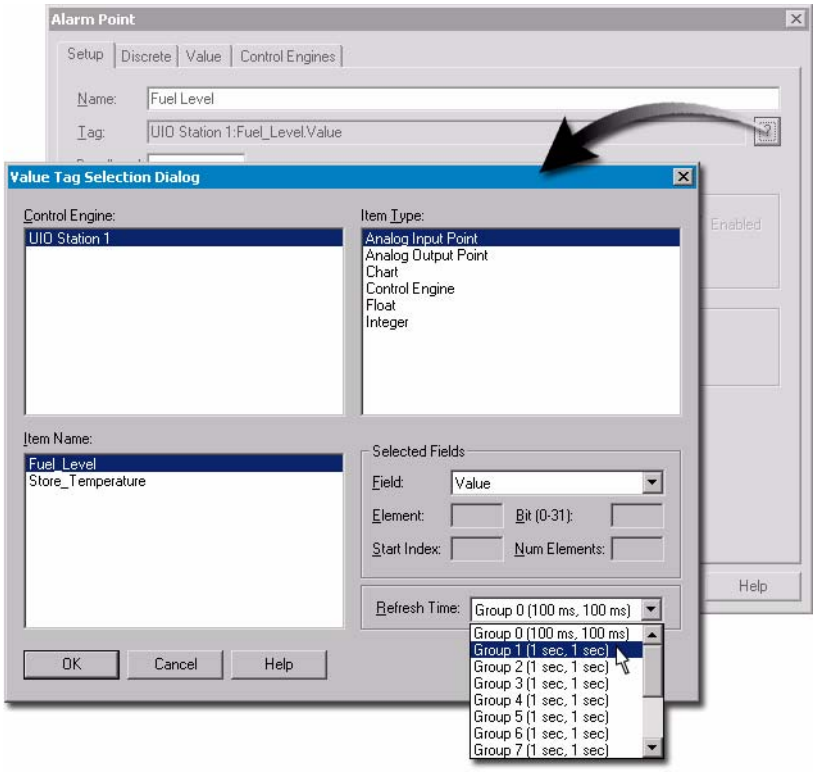
b. Type `Fuel Level` in the Name field.

c. Under *Setup By*, choose *Current Value*. (It should be the default selection.)

NOTE: It is important that you select the correct *Setup By* option before selecting a tag, as the *Setup By* option determines which tags will be available.

3. Choose the tag.

a. Click the Tag Selection button to open the Value Tag Selection dialog box.



b. In the Discrete Tag Selection dialog box, select the following:

Control Engine: *UIO Station 1*.

Item Type: *Analog Input Point.*

Item Name: *Fuel_Level.*

Field: *Value.*

Refresh Time: *Group 1.*

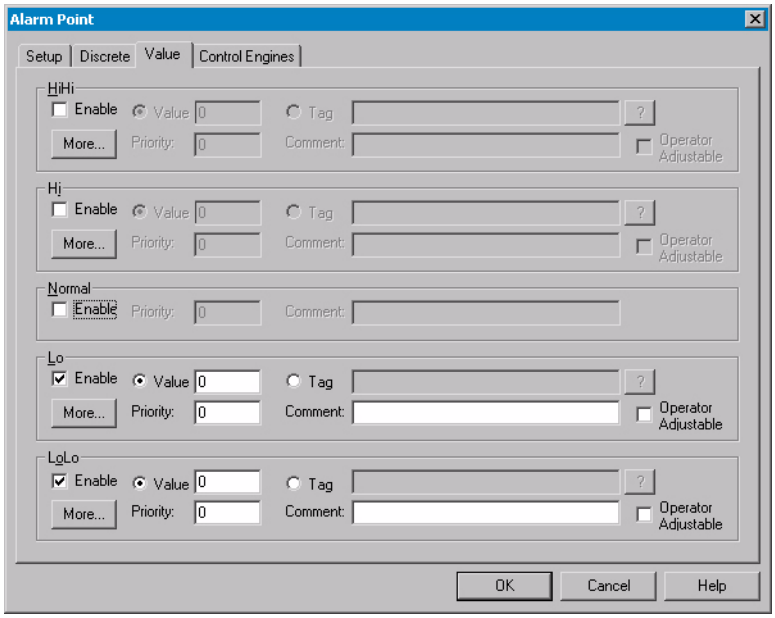
c. Click *OK* to close the Value Tag Selection dialog box.

4. Configure alarm values.

You will configure this alarm to react to Lo and LoLo regular fuel levels.

a. Click the *Value* tab.

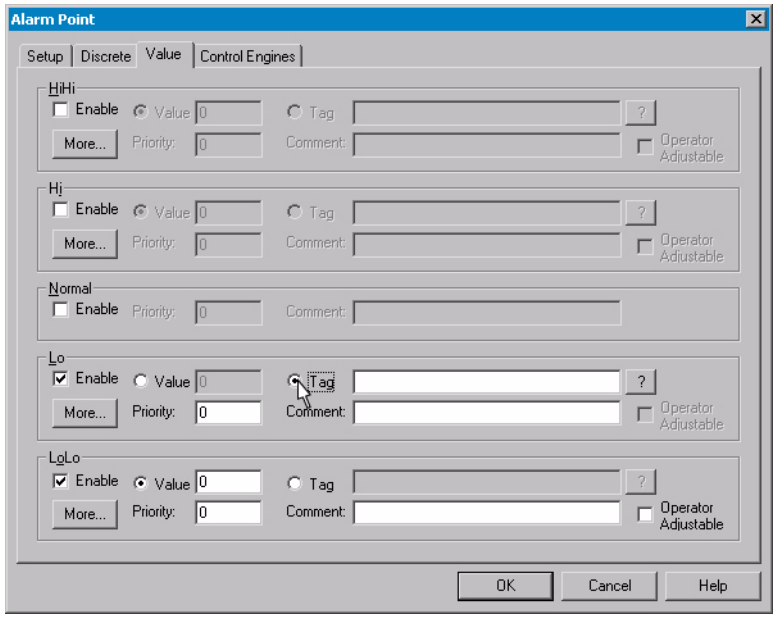
b. Click the *HiHi*, *Hi*, and *Normal* check boxes to disable alarms for these states.



5. Assign a tag to the alarm criteria.

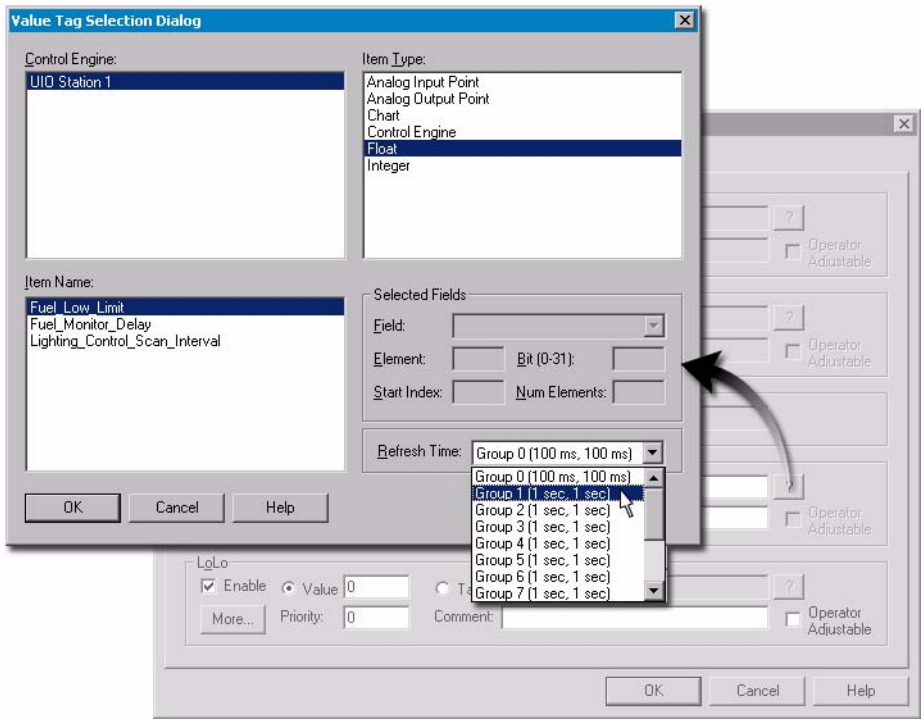
The Lo alarm will be determined by the value of Fuel_Low_Limit, rather than a fixed value.

a. In the *Lo* field, select the *Tag* option.



NOTE: Selecting a tag allows you to evaluate an alarm state based on a variable.

b. Click the *Tag Selection* button.



c. In the Value Tag Selection dialog box, select the following:

Control Engine: Choose *UIO Station 1*.

Item Type: Choose *Float.*

Item Name: Choose *Fuel_Low_Limit.*

Refresh Time: Choose *Group 1.*

d. Click *OK* to close the Value Tag Selection dialog box.

6. Configure Lo Priority value and comment.

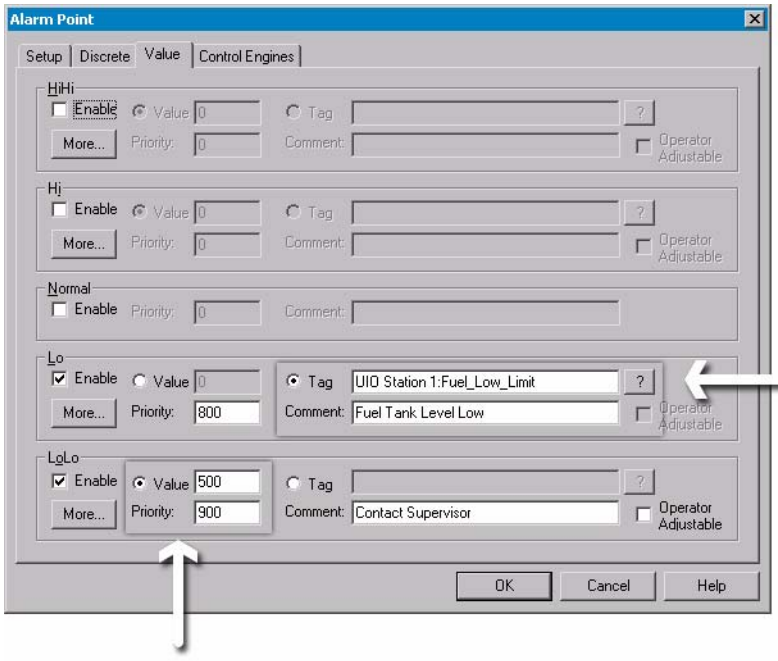
You can assign a priority to your alarm point on a scale of 0–999. The lowest priority is 0, and the highest is 999. (Later, you will see how an operator can filter alarms by their priority settings.)

a. Type `800` in the Priority field.
b. Type `Fuel Tank Level Low` in the Comment field.

7. Add value, comments, and priority to LoLo value.

a. Type `500` in the *Value* field.
b. Type `900` in the *Priority* field
c. Type `Contact Supervisor` in the *Comment* field.
d. Close the Alarm Point dialog box.

Literal and Tag-based Alarm Criteria:



Lo alarm triggered by comparison to another tag. This value can be changed.

LoLo alarm triggered by comparison to value entered here. This value is fixed.

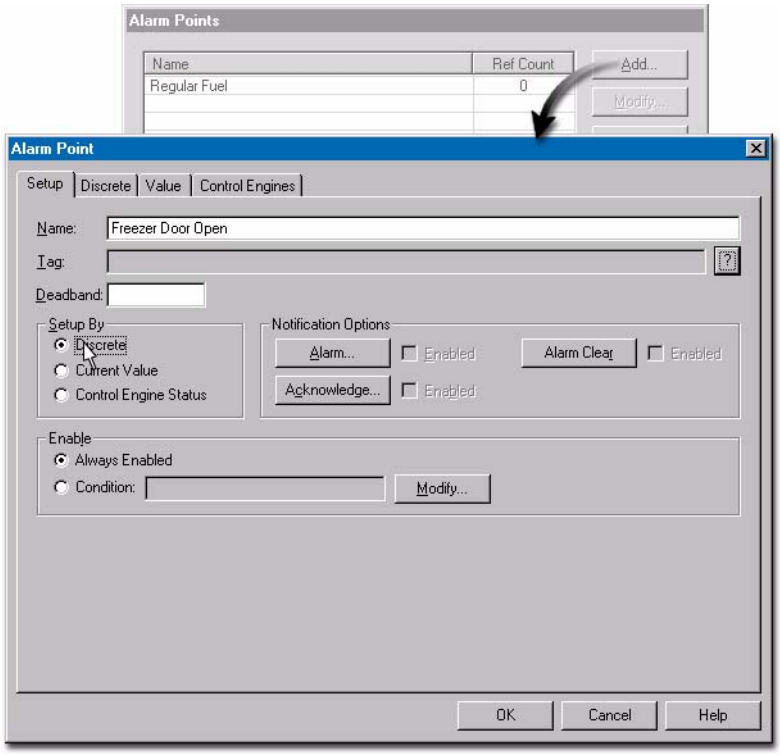
Configuring the Freezer Door Alarm Point

The next alarm point will be a discrete item. We want to be notified if the freezer door is open for more the 10 seconds.

1. Create Freezer Door Alarm Point.

a. In the Alarm Points dialog box, click *Add* to open the Alarm Point dialog box.

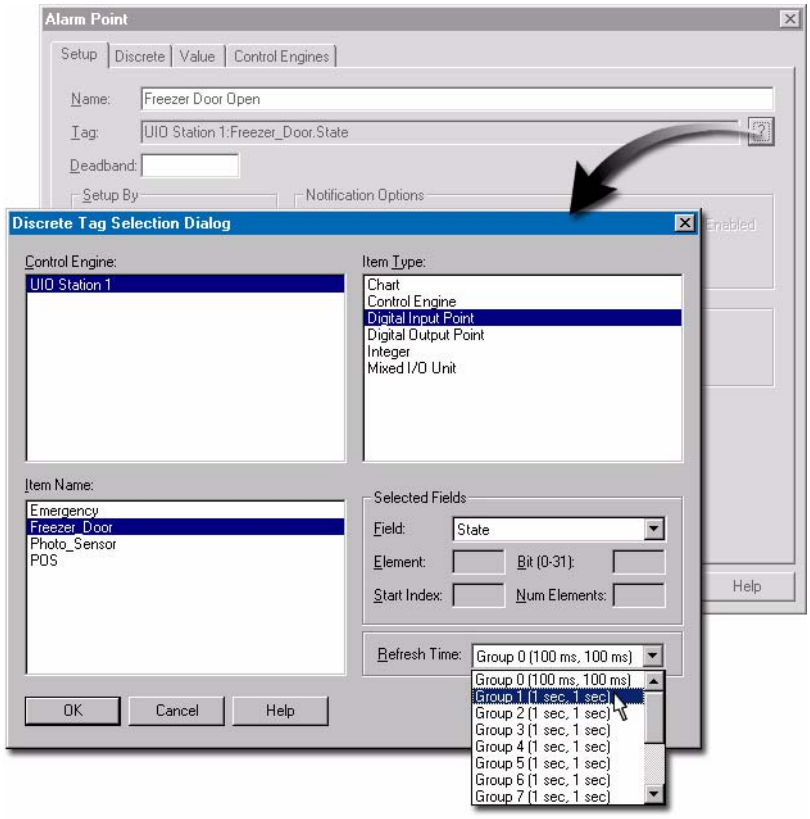
b. Type `Freezer Door Open` as the name of this alarm point.



c. Under *Setup By*, click *Discrete*.

2. Choose Tag.

a. Click the *Tag Selection* button.



b. In the Value Tag Selection dialog box, select the following:
Control Engine: *UIO Station 1*.

Item Type: *Digital Input Point.*

Item Name: *Freezer_Door.*

Field: *State*.

Refresh Time: *Group 1.*

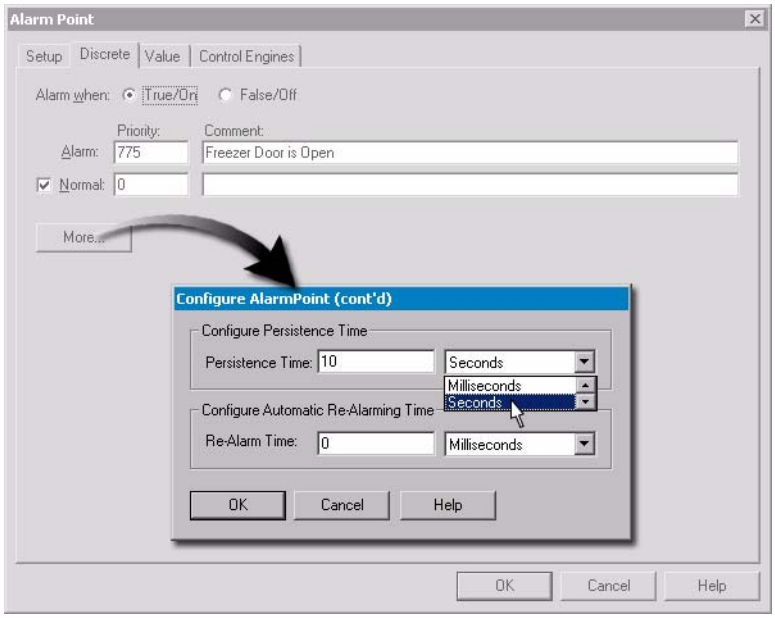
c. Click *OK* to close the Value Tag Selection dialog box.

3. Discrete Tag Options.

a. Click the *Discrete* tab.
b. For *Alarm when*, choose *True/On*.

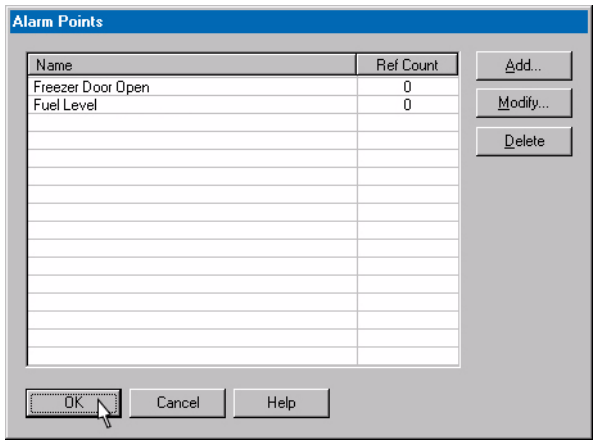
This will put the Freezer Door in an alarm state when it is open. (You can configure a discrete alarm state as either On or Off.)

c. In the *Priority* field, type `775`.
d. In the Comment field, type `Freezer Door is Open`.
e. Click the *More* button.



f. Under Configure Persistence Time, enter *10 Seconds*.
g. Click *OK*.
h. Click *OK* to close the Alarm Point dialog box.

4. Close the Alarm Points dialog box.



Creating a New Window for Alarm Graphics

Alarms are displayed within alarm graphics that you place in a window. In this section, you will create a new window to hold the alarm graphics. The window will be configured to open automatically when an alarm occurs and to require the operator to acknowledge the alarm before closing the window.

1. Open the alarm window configuration options.

a. Select *Window* ⟶ *New*.
b. Type `Alarms` in the Name field.
c. Type `0` in both the X Position and Y Position fields.
d. Type `800` in the Width field and `600` in the Height field.
e. Select *System Menu*.

2. Select runtime options.

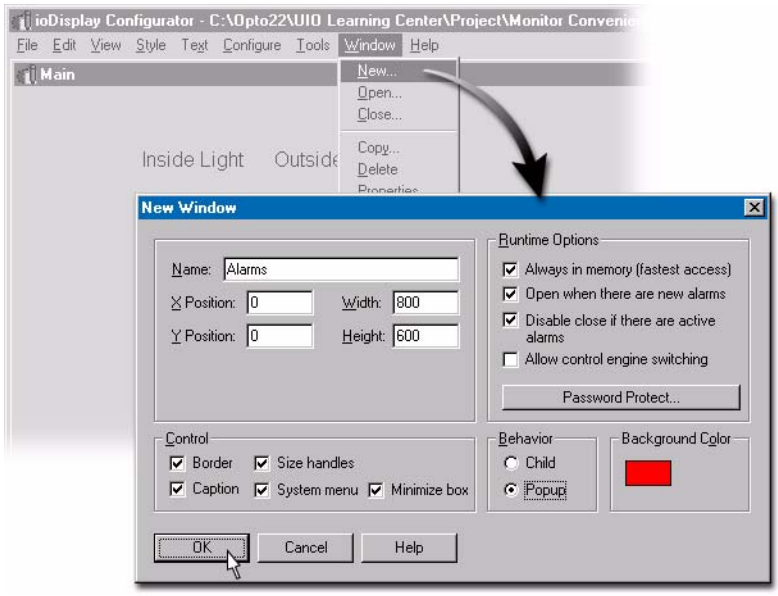
a. Check the following boxes in Runtime Options:

Always in memory

Open when there are new alarms

Disable close if there are active alarms

b. Select *Popup* as the Behavior. (This means that the window will stay on top of all windows, even if the operator clicks on a window behind it.)
c. Select Red as the Background Color.



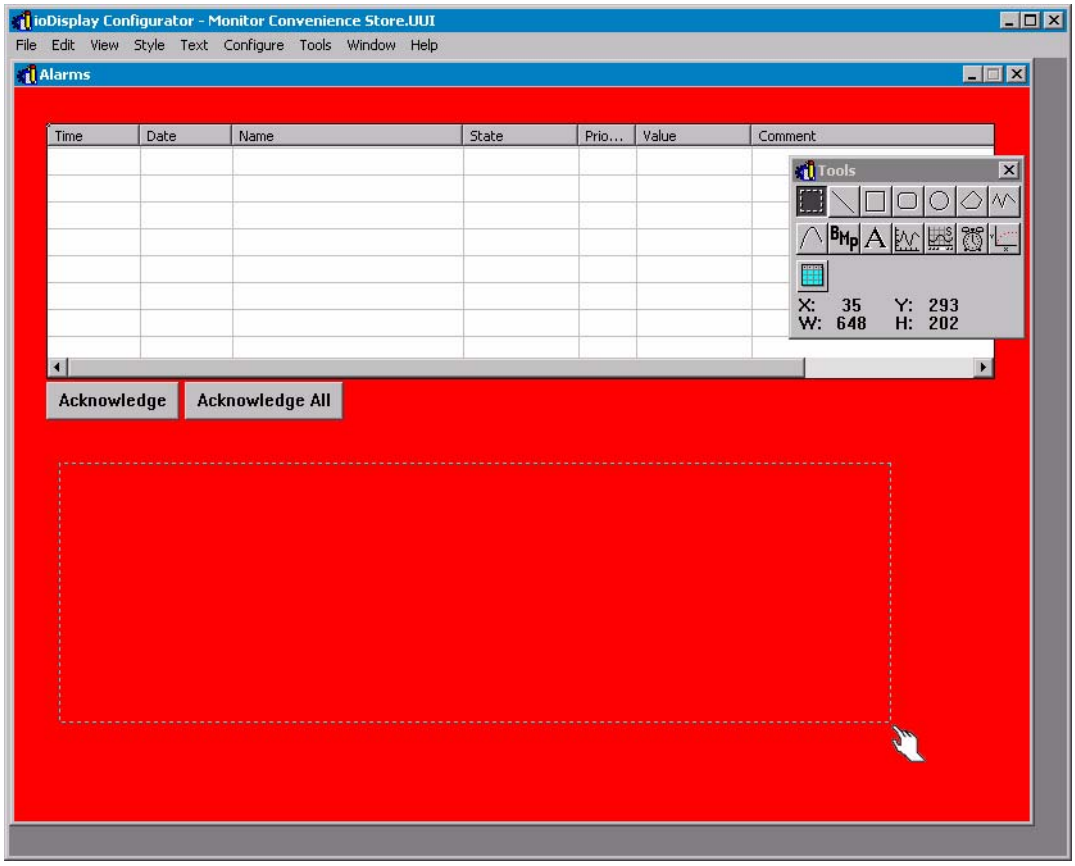
d. Click *OK* to close the New Window dialog box.

Creating a Detailed Alarm Graphic

You will create two alarm graphics in the Alarms window. One graphic will be a Detailed alarm, and the other will be a History alarm.

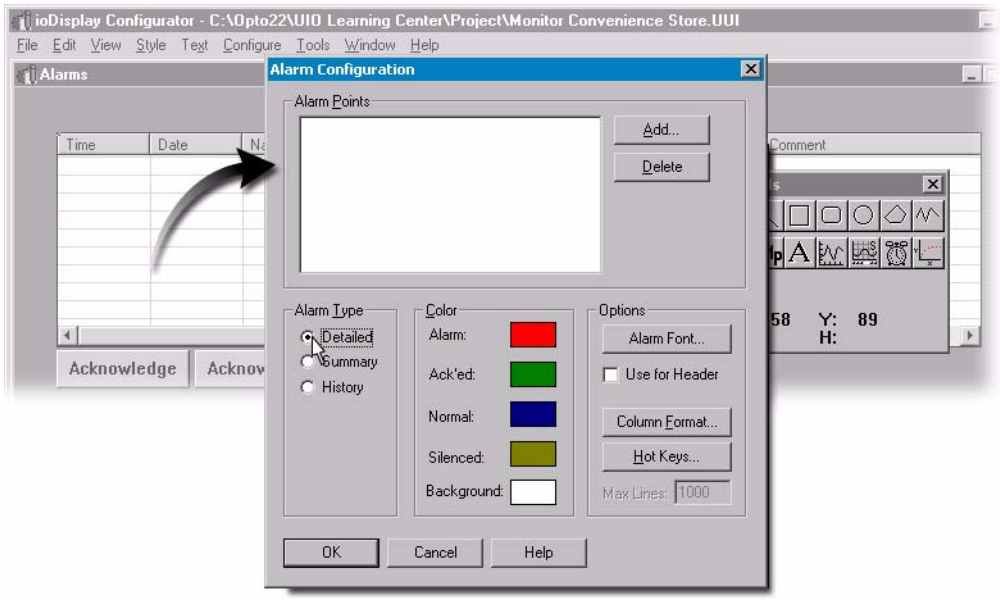
1. Draw alarm graphics.

a. Select the Alarm tool from the Toolbox.
b. Starting in the upper left-hand corner of the *Alarms* window, click and drag a large rectangle. The rectangle should be approximately half the vertical size of the *Alarms* window.
c. With the Alarm tool still selected, click and drag another rectangle in the lower half of the window, to create the second alarm graphic.



2. Configure the alarm graphic.

a. Choose the Select tool.
b. Double-click the first alarm graphic.



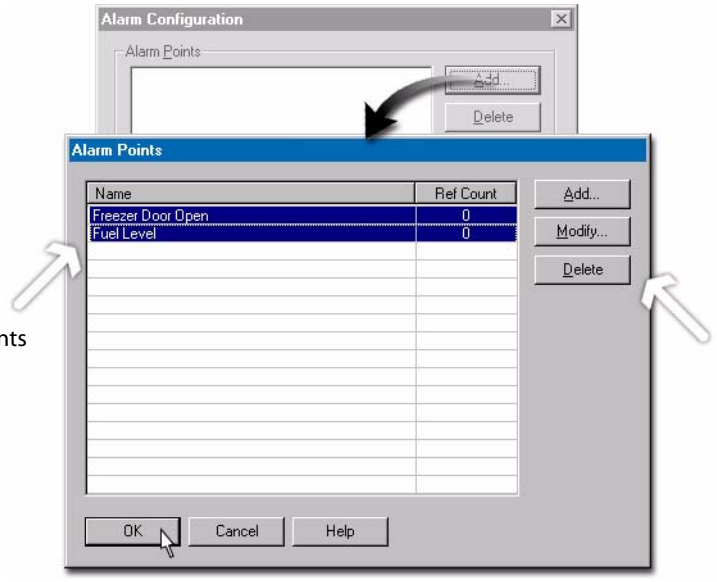
c. Select *Detailed* as the *Alarm Type*.

Detailed alarms treat each alarm point state as a separate alarm condition. Each alarm point state needs to be acknowledged before it is removed from the alarm graphic. Therefore, if an alarm goes from Hi to HiHi, both alarm states will be listed. (Summary alarms display only the current state of an alarm point.)

3. Add alarm points to the detailed alarm graphic.

a. In the Alarm Configuration dialog box, click *Add*.
b. Select both alarm points by clicking one and then shift-clicking the other.

Selecting alarm points will include them in your alarm graphic.

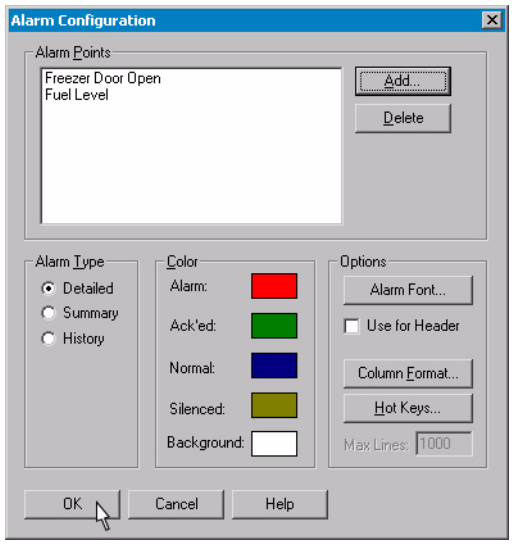


NOTE: Be careful with the *Add*, *Modify*, and *Delete* buttons. *Add* and *Modify* take you to the Alarm Point dialog box to add or modify the alarm point. *Delete* will remove alarm points from your ioDisplay project.

c. Click *OK* to close the Alarm Points dialog box.

4. Close the Alarm Configuration dialog box.

Click *OK*.



5. Adjust columns.

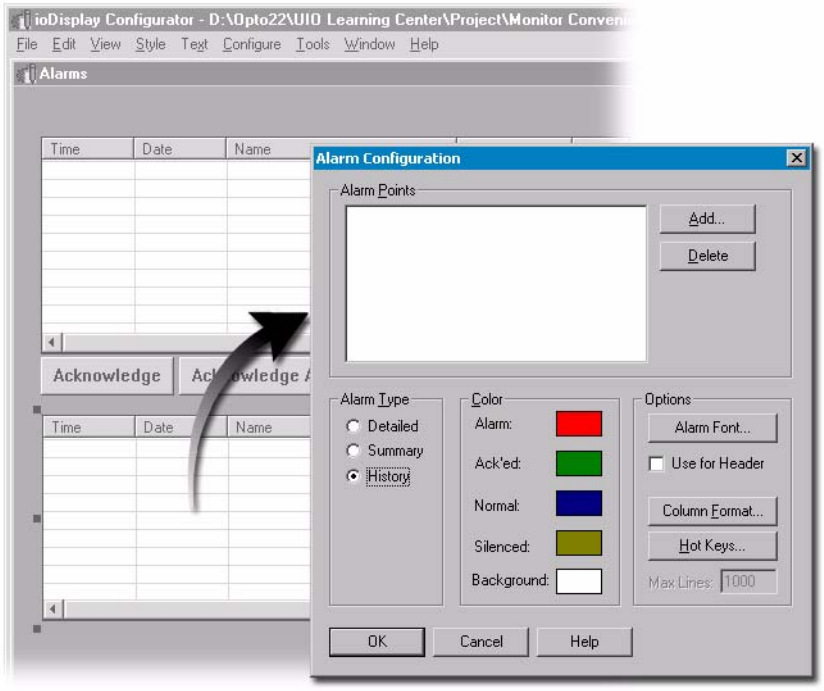
a. To resize a column, move the cursor to the right of the field containing that column's caption.

b. When the double-headed arrow appears, click and drag your mouse to the desired width.

Creating a History Alarm Graphic

1. Open the Alarm Configuration dialog box.

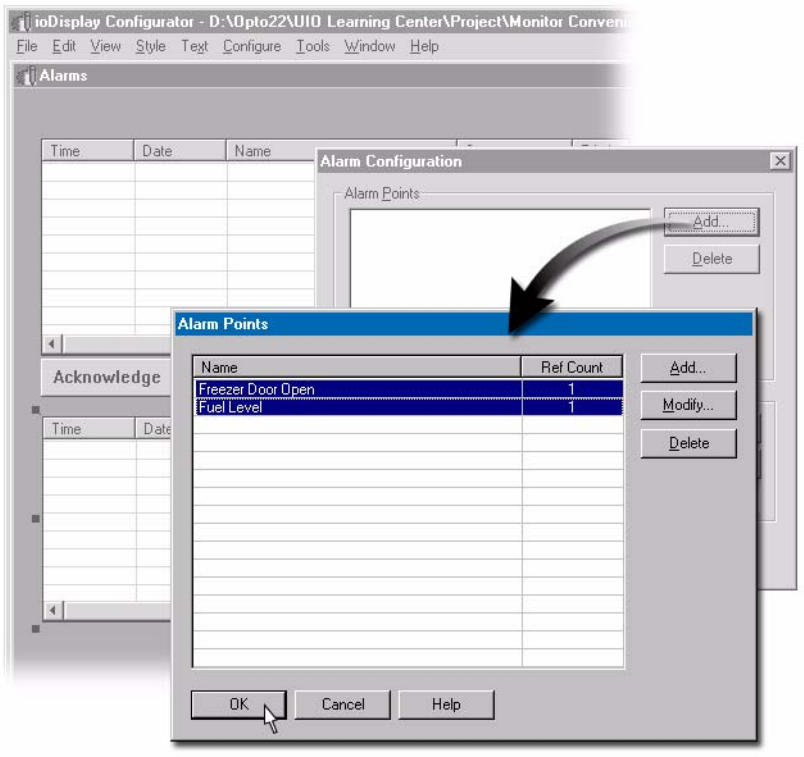
a. Double-click the lower alarm graphic.
b. Select *History* as the *Alarm Type*.



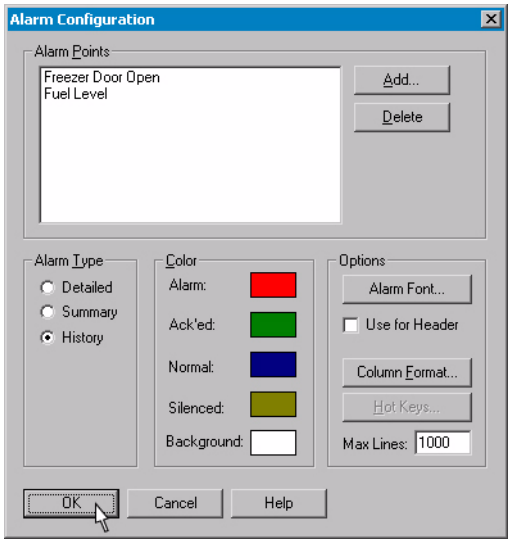
This graphic will display a complete list of all alarms and their status.

2. Add Alarm Points.

a. In the Alarm Configuration dialog box, click *Add*.
b. Select both alarm points by clicking one and then shift-clicking the other.



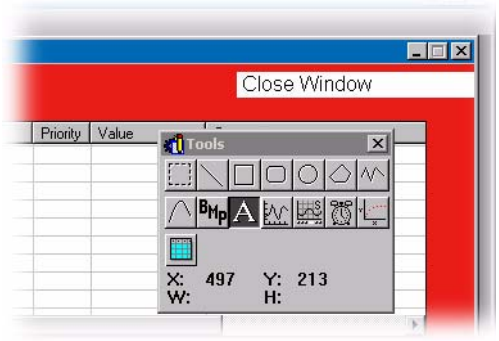
c. Click *OK* to close the Alarm Points dialog box.
d. Click *OK* to close the Alarm Configuration dialog box.



Creating a Close Window Button

In this section you will create text that will act as a button to close the Alarms window. Creating a button involves drawing the object that will be the button and then assigning it an operator-driven attribute.

1. Create text.



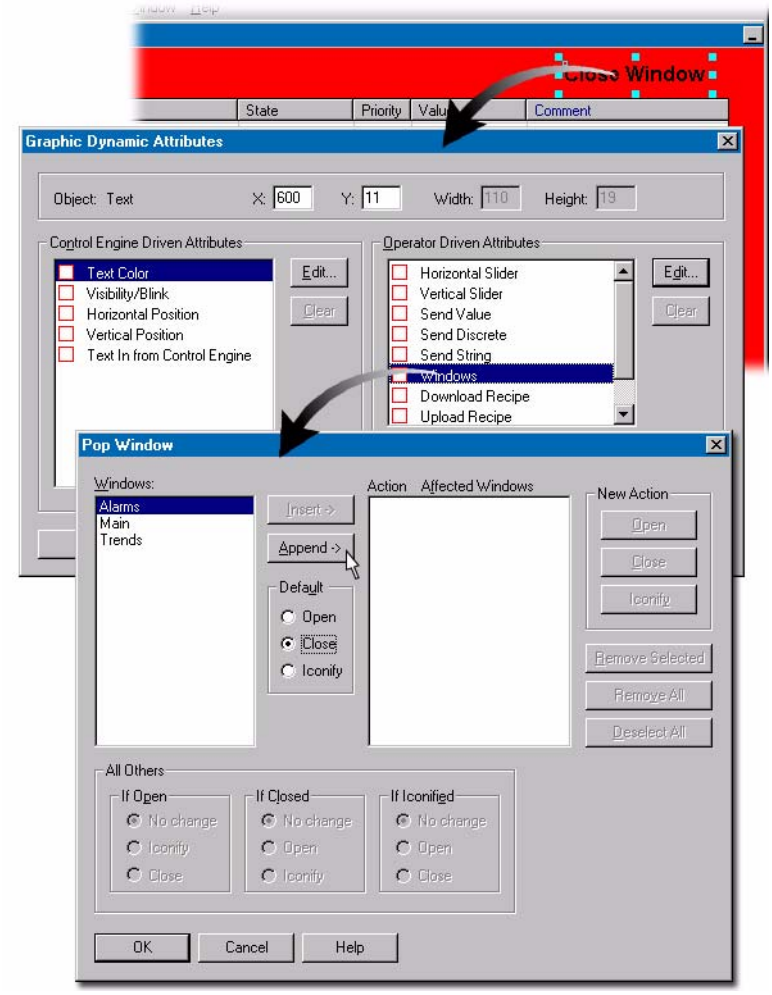
a. Select the Text tool from the Tools palette.
b. Click in an empty space at the top right of the Alarms window.
c. Type `Close Window`.
d. Click elsewhere to anchor the text.

2. Format the text.

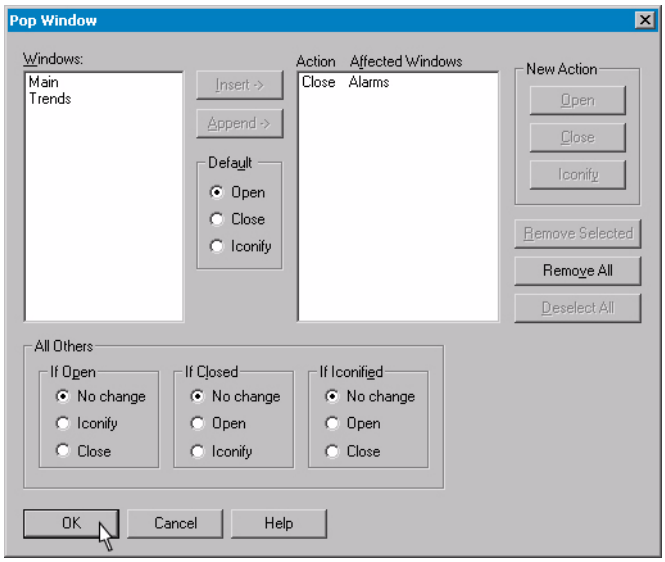
a. Choose the Select tool from the Toolbox.
b. Right-click on the *Close Window* text.
c. From the shortcut menu select *Style* ⟶ *Bold*.

3. Assign operator-driven attributes.

a. Double-click the *Close Window* text to open the Graphic Dynamic Attributes dialog box.
b. Double-click *Windows*.
c. Choose *Alarms* in the Windows field.
d. Select *Close* in the Default field.
e. Click *Append*.



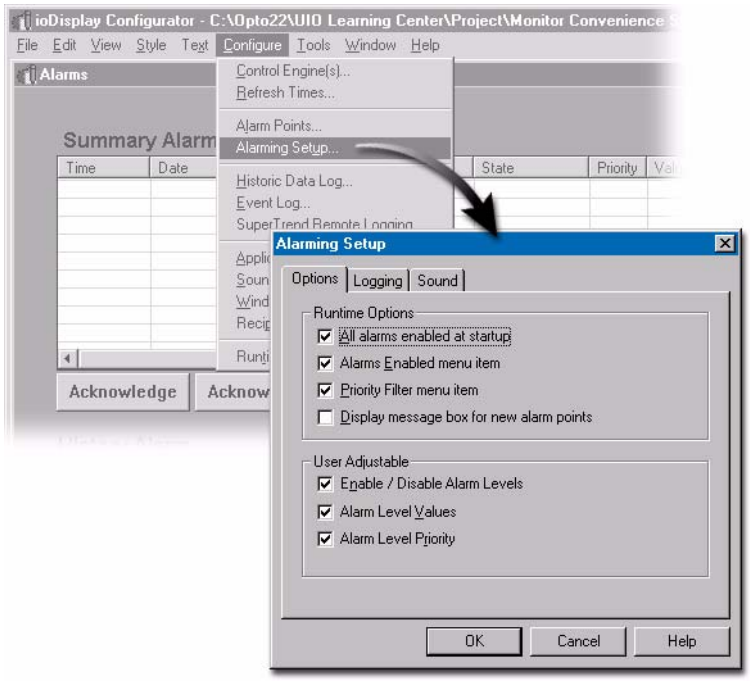
b. Click *OK* to close the Pop Window dialog box.
c. Click *OK* to close the Graphic Dynamic Attributes dialog box.



Examining Alarming Setup Features

1. Configure Alarming Setup.

a. Select *Configure* ➞ *Alarming Setup*.



b. Under *Runtime Options*, select the following:

All alarms enabled at startup

Alarms Enabled menu item

Priority Filter menu item

c. Check all items under *User Adjustable*.

About Runtime Options

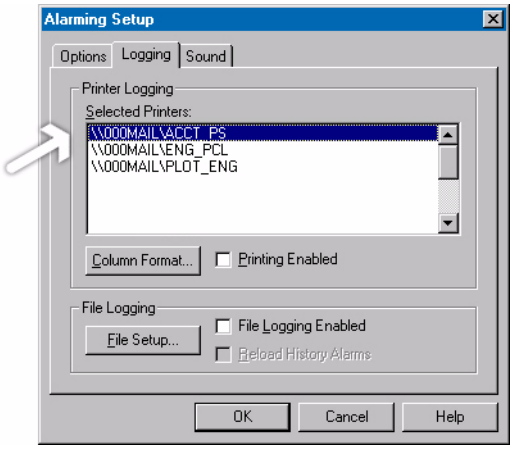
From the *Options* page, you can select several runtime settings. If *All alarms enabled at startup* is checked, all alarms will be enabled as soon as ioDisplay Runtime connects to the control engine. If *Alarms Enabled menu item* is checked, the user has access to the *Alarms Enabled* menu item at Runtime. By selecting this menu item, the user can enable or disable all alarms. If this item is not checked, the menu item will be grayed out and inaccessible. If *Priority Filter menu item* is checked, the user has access to the *Priority Filter* menu item at Runtime. By selecting this menu item, the user can filter out alarms lower than a specified priority. If this item is not checked, the menu item will be grayed out and inaccessible. If *Display message box for new alarm points* is checked, an Alarm Point Message dialog box is displayed for each alarm point. This message box displays information about the alarm point. The alarm can be acknowledged from this dialog box, or the dialog box can be closed and the alarm acknowledged later. You can also give users the ability to change alarm levels and priorities at Runtime.

2. Examine alarm logging options.

From the Logging page, you can configure printer and file alarm logging.

a. Click the *Logging* tab.

Highlighting indicates a selected printer.



About Printer Logging

The *Selected Printers* field displays all installed printers. Multiple printers may be selected to log alarms. The *Column Format* button displays the Alarm Format dialog box. From this dialog box, the printed columns can be configured. Each column is optional and the width can be specified. The *Printing Enabled* check box enables printer logging. The *File Setup* button displays the File Access dialog box, used to configure the log file's file name, line format, and rollover period. The *File Logging Enabled* check box enables file logging.

Any printer configured on your PC is displayed in the Selected Printers dialog box.

b. If you wish to experiment with the print logging feature, select a printer and check *Printing Enabled*.

You can send alarms to additional printers by selecting more than one printer in the Selected Printers field.

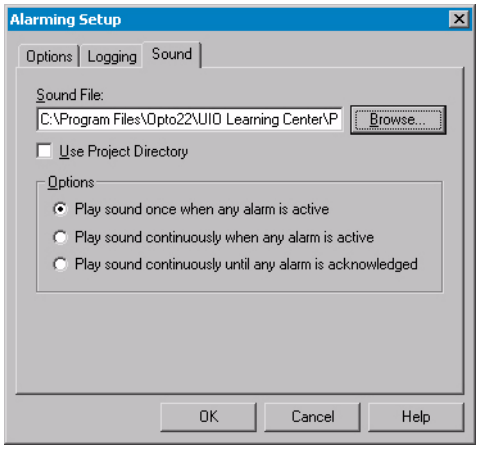
3. Create an auditory alarm.

The *Sound* page configures the alarm sound capabilities. To use these features, your PC must have a properly configured sound card and driver to support the desired sound file types. Both WAV and MID files are supported.

a. Click the *Sound* tab.

b. Click *Browse* to choose the sound file to play.

(If the *Use Project Directory* option is selected, only the file name needs to be specified. If this option is not selected, the path must be included with the file name.)



c. Move to the **C:\Program Files\Opto22\UIO Learning Center\Project** directory if necessary.

d. Double-click the `Ahhhhh.wav`.

e. Make sure the option *Play sound once when any alarm is active* is selected.

f. Click *OK*.

Testing Alarms

1. Correct any alarm conditions.

a. Turn the potentiometer so the tank level is above 2000 gallons.
b. Turn off the Freezer Door switch (Freezer Door Status light should be off).

2. Launch ioDisplay Runtime.

a. Select *File* ➞ *Save Project and Load Runtime*.
b. Close the Event Log Viewer.

3. Trigger alarms.

a. Slowly turn the Fuel Level potentiometer counter-clockwise, to decrease the tank level.

   When it reaches the value designated as the Lo alarm (the value of Fuel_Low_Limit, set to 1000 by default), the alarm is recorded in both the summary alarm (the top graphic) and the history alarm (the lower graphic).

b. Continue to decrease the Fuel Level, until the LoLo alarm is triggered.
c. Turn the Fuel Level potentiometer clockwise to increase the tank level so that alarms for this point are no longer triggered.

   Note that the historic log records when the tank level returns to a "Normal" level.

d. Click the *Close Alarms Window*.

   Nothing happens. When you configured the Alarms window, you checked an option box that prevents the window from closing while there are active alarms.

e. Click the *Acknowledge All* button. Notice that the alarms change colors when they are acknowledged.

4. Filter alarms.

a. From the *Alarm* menu, select *Priority Filter*, and then *900 And Above*.

   With this option selected, only alarms with a priority of 900 or greater will be displayed. Note that from the *Alarm* menu you can also disable all alarms. If you wish to prevent a user from enabling/disabling alarms, you can disable this feature in the Configurator.

b. Turn the Fuel Level potentiometer counter-clockwise to trigger the Lo and LoLo alarm.

   With the alarm filter set at 900 and above, only the LoLo alarm is displayed.

c. From the *Alarm* menu, select *Priority Filter*, and then *All Priority Levels*. This will enable alarms of any priority level.
d. Turn on the Freezer Door switch and wait 10 seconds.

   An alarm will appear in the alarm summary.

e. Click the *Freezer_Door* alarm to select it.
f. Turn off the Freezer Door switch.
g. Click *Acknowledge*.
h. Click the *Acknowledge All* button to clear any alarms.
i. Click *Close Alarms Window*.

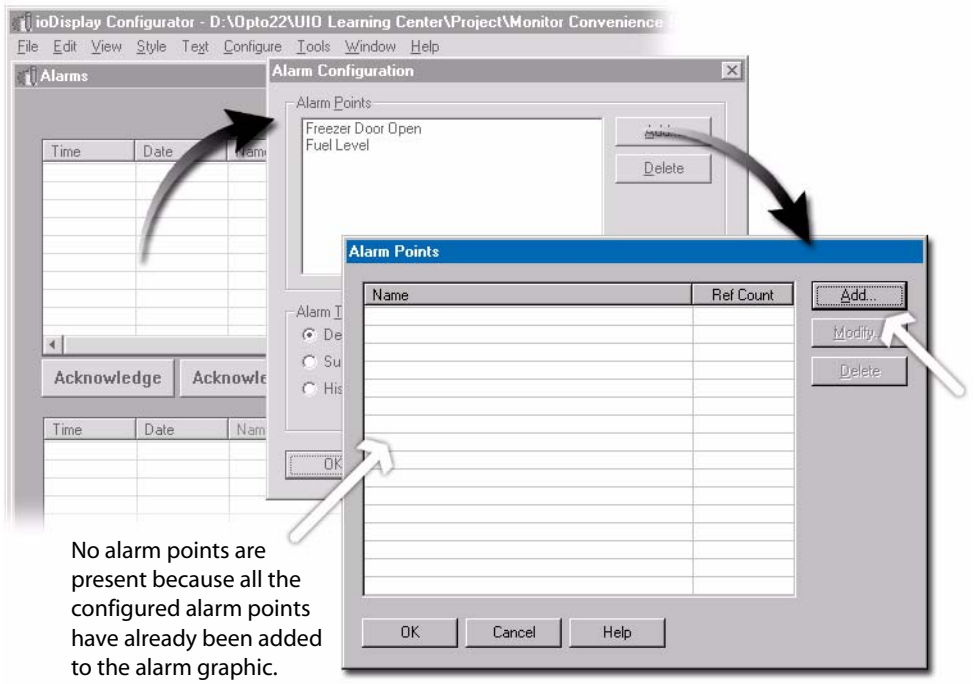
5. Exit ioDisplay Runtime.

Configuring Control Engine Alarming

The previous alarms were based on the state of I/O points. But what if your control engine loses power, or gets disconnected from the Ethernet hub, or someone temporarily takes the control engine off-line to update the ioControl strategy? You might want your ioDisplay operator interface to notify you of such events. To create this ability, you need to create a control engine alarm point and then add it to your alarm graphics.

1. Add a control engine alarm point to the detailed alarm.

a. Double-click the upper alarm graphic.
b. In the Alarm Configuration dialog box, click *Add*.

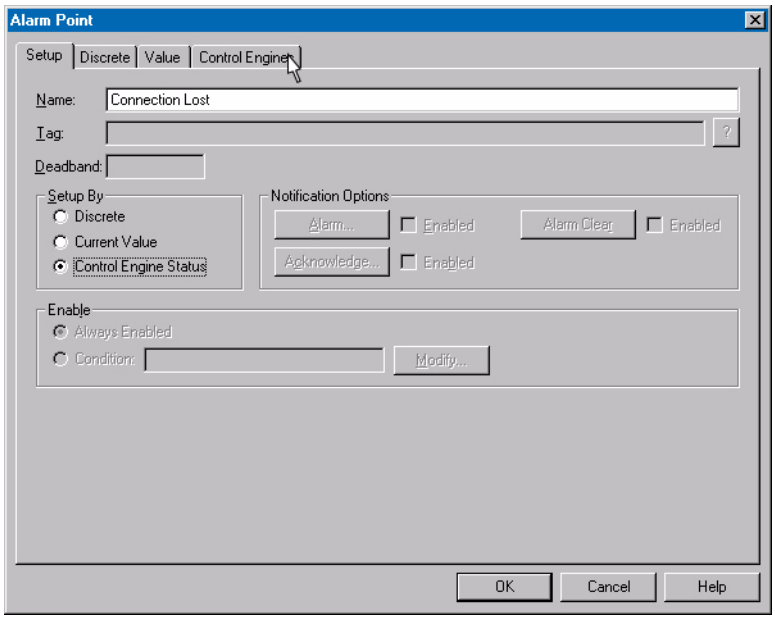


c. In the Alarm Points dialog box, Click *Add*.

   This time, you are configuring an alarm point on-the-fly. (You could have first added the alarm point using the *Configure* ⟶ *Alarm Points* menu option, and then added the alarm point to the alarm graphic.)

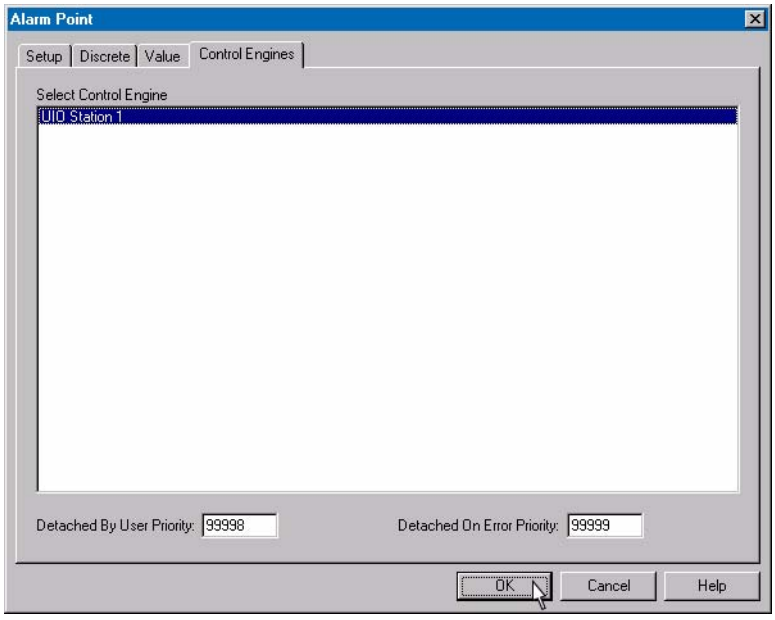
d. In the *Name* field, type `Connection Lost`.

e. In the *Setup By* field, select the *Control Engine Status* option.



2. Choose a control engine.

a. Click the *Control Engines* tab.
b. Select *UIO Station 1.*



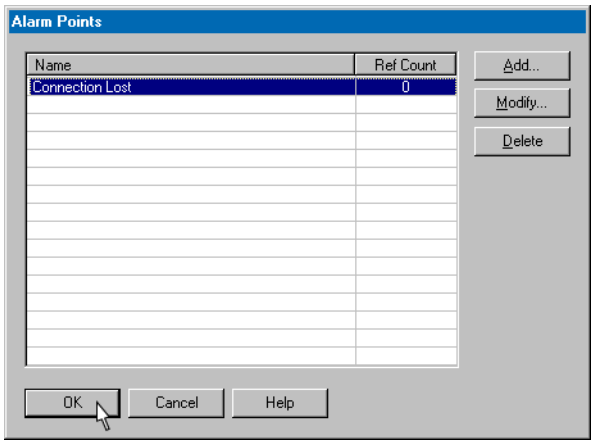
NOTE: If your control engine isn't shown, make sure Control Engine Status, under Setup, is selected.

Any control engines that have been added to the ioDisplay project are shown here. If you want to change the Detached by User or Detached on Error priority levels, enter a new value in the corresponding field. Using the default settings, these priority levels cannot be filtered out by the user since they are higher than 999. If you want the user to be able to filter out control engine status alarms, set either—or both—values to 999 or less.

c. Click *OK* to close the Alarm Point dialog box.

3. Add a new alarm point.

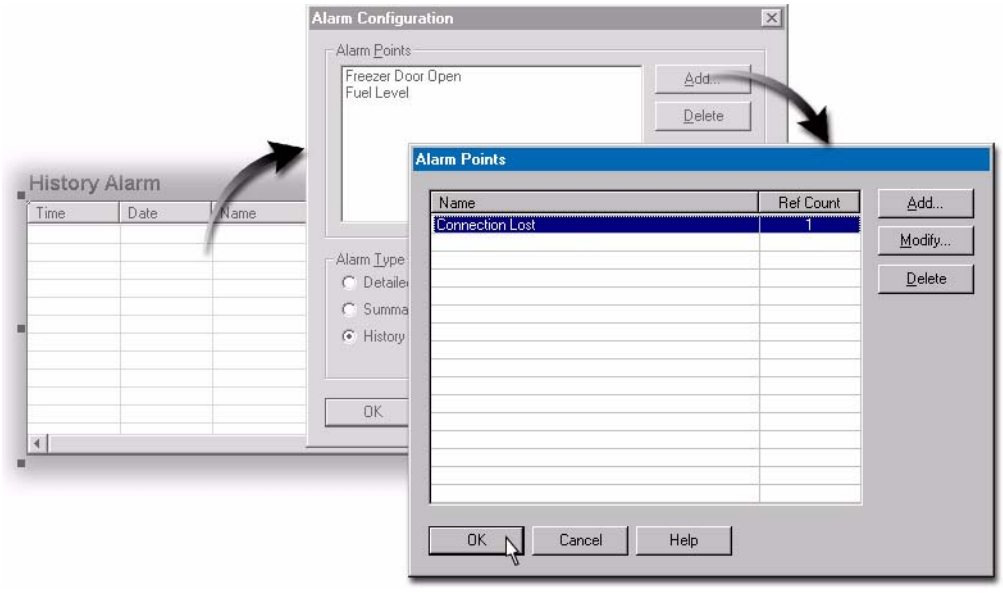
a. In the Alarm Points dialog box, select *Connection Lost*.



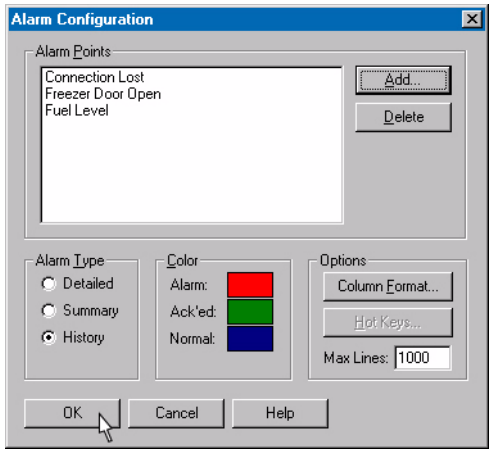
b. Click *OK* to close the Alarm Points dialog box.
c. Click *OK* to close the Alarm Configuration dialog box.

4. Add the new alarm point to the history alarm.

a. Double-click the lower alarm graphic.
b. In the Alarm Configuration dialog box, click *Add*.
c. Select *Connection Lost*.



d. Click *OK* to close the Alarm Points dialog box.



e. Click *OK* to close the Alarm Configuration dialog box.

Configuring a Dynamic Object for Control Engine Status

1. Open the Main window.

2. Create a graphic.

You will create a graphic that will change color based on the status of the control engine connection. This graphic will also launch the View Control Engine dialog box that lets you attach or detach the control engine connection.

a. Choose the Rectangle tool.

Review: Opening a closed window

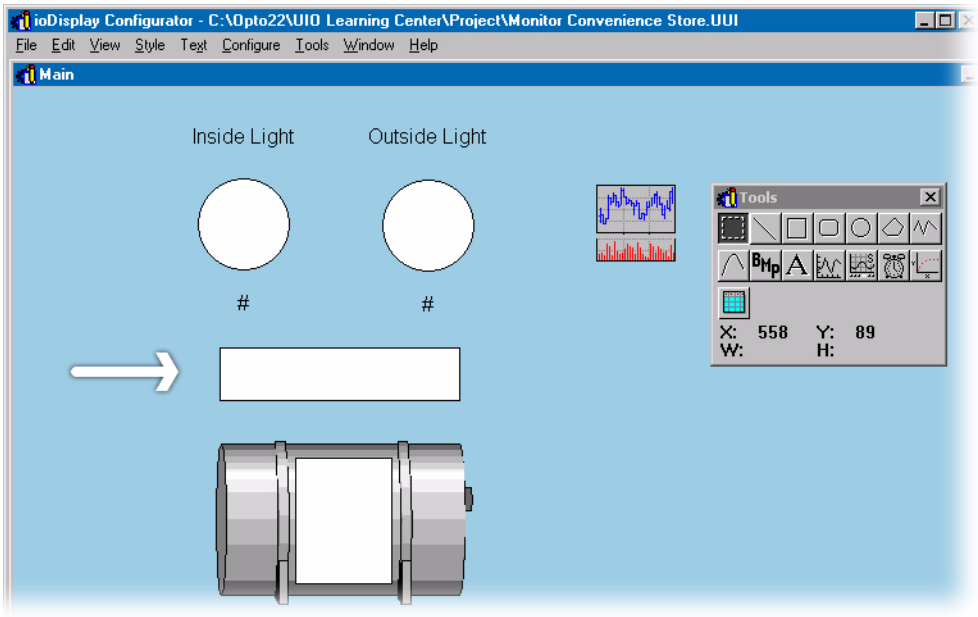
1. Choose *Window* ⟶ *Open*.

(This option is disabled when all windows are already open)

2. In the Open Windows dialog box, choose the window.

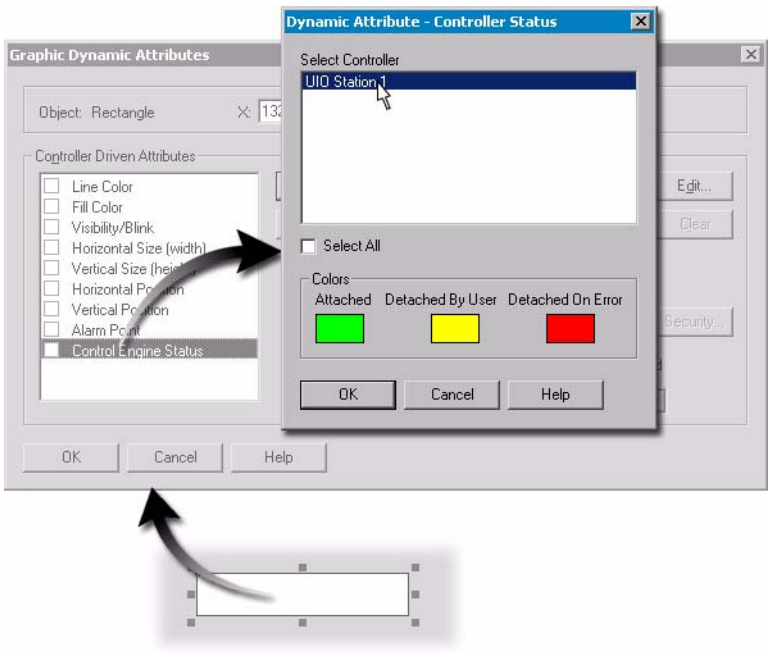
3. Click *OK*.

b. Click and drag a rectangle in the window:



3. Assign control engine-driven attributes.

a. Choose the Select tool.
b. Double-click the rectangle you just created.
c. In the Graphic Dynamic Attributes dialog box, double-click *Control Engine Status*.
d. In the Dynamic Attribute dialog box, select *UIO Station 1*.



e. Change the color fields if desired. By default, the button will be light green if the control engine is attached, yellow if the control engine is detached by the user, red if the control engine is detached by an error, and the dynamic objects will be dark green to indicate a value is the last known value before the control engine connection was lost.

Review: Changing a Color

Click a color box to open the Colors dialog box. In the Color dialog box, you can select an existing color or define a new one. Open the Color dialog box for each color you configure.

f. Click *OK* to close the Dynamic Attribute dialog box.

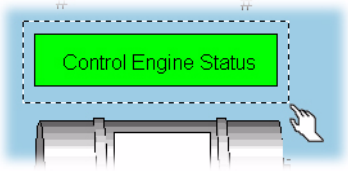
Selecting Control Engine Status disables the other control engine-driven attributes. This graphic now reflects the Control Engine Status, which is either Attached, Detached by User, or Detached on Error.

4. Create a text label.

a. Choose the Text tool.
b. Click in the middle of the rectangle.
c. Type `Control Engine Status`, and click elsewhere in the window to anchor the text.

5. Format the text.

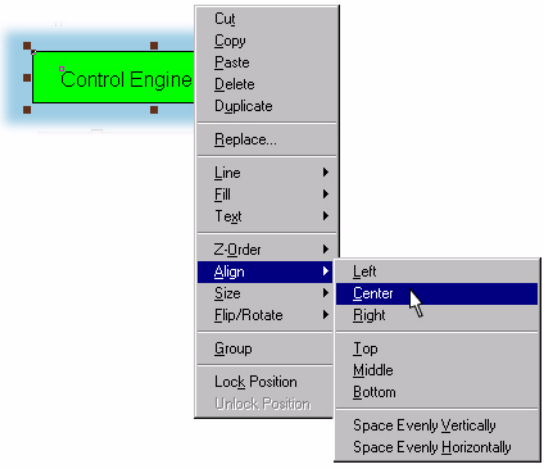
a. Click the Select tool.
b. Select the text.
c. Right-click the text and choose *Style* ⟶ *Bold* from the pop-up menu.
d. Select both the text and the rectangle.



Review: Selecting Multiple Objects

With the select tool, click and drag a rectangular area around the objects you wish to select.

e. Right-click the selected objects and choose *Align* ⟶ *Center* from the pop-up menu.



f. Right-click the selected objects again, and choose *Align* ⟶ *Middle*.

Aligning these objects may have moved them.

g. Drag the selected objects or use the arrow keys to move these objects to the desired location in your window.

Testing Your Project

1. Correct any alarm conditions.

a. Turn the potentiometer so the tank level is above 1000 gallons.
b. Turn off the Freezer Door switch (Freezer Door Status light should be off).

2. Launch ioDisplay Runtime.

a. Select *File ➞ Save Project and Load Runtime*.
b. Click *Close* to close the Event Log Viewer.

3. Simulate a control engine error.

To simulate a control engine error, disconnect the Ethernet cable from the SNAP Ultimate brain.

Three things will happen:

- The Event Log Viewer notifies you that the control engine has detached on error.
- Alarms appear in both alarm graphics.
- The dynamic objects turn dark green to indicate that their values are the last known value. (You will need to move the Alarms window to see this.)

4. Restore connection.

Reconnect the Ethernet cable to the SNAP Ultimate brain.

ioDisplay will automatically restore communications. After a minute or so, three things will happen:

- The Event Log View notifies you that the control engine has reattached.
- Messages appear in both alarm graphics.
- The dynamic objects return to their normal colors.

5. Close the Alarm Window.

a. Acknowledge any remaining alarms.
b. Click *Close Window.*

6. Close ioDisplay Runtime.

7. Close ioDisplay Configurator.

SUMMARY

Dynamic graphics, SuperTrends, and Alarms are three commonly used components of ioDisplay. There are many other features to discover on your own:

- **Additional dynamic attributes**—You haven't used all of the dynamic attributes available. You may wish to experiment with the operator-driven attributes that manipulate an objects visibility and position on the screen. These features are

useful for creating elaborate animation using graphics drawn in ioDisplay or graphics created in other illustration programs.

- **Additional graphical objects**—You may wish to use table graphics to display the values of numerical table variables used in your ioControl strategies.
- **Recipe files**—Text-based recipe files can write to variables in your ioControl strategy. The capabilities of the operator's interface are greatly extended by creating recipe files that can be downloaded from ioDisplay.
- **System architecture options**—In an industrial network, you may wish to use a dedicated server for data management and storage of SuperTrend data. You can designate remote computers for these tasks.

This last option includes using OptoOPCServer running on a dedicated server. This configuration is recommended for applications where multiple seats of ioDisplay are monitoring a process. OptoOPCServer is described in the next lesson.

OPTOOPCSERVER

SKILLS

OptoOPCServer

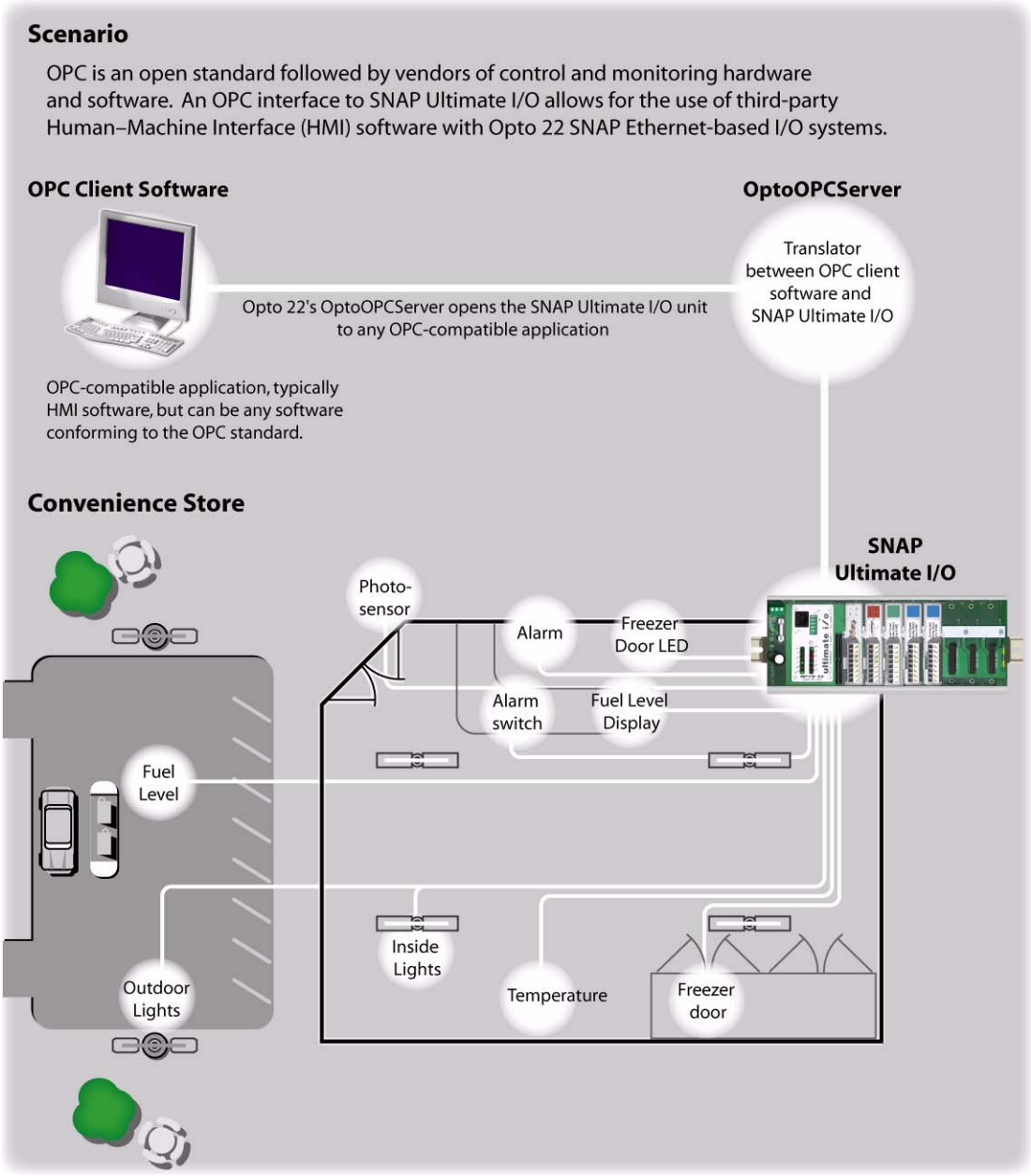
- Configuring OptoOPCServer
- Connecting a test OPC client to OptoOPCServer
- Creating a tag group
- Configuring tags

OPC Test Client

- Monitoring point values and states
- Writing to a point

SCENARIO

In this lesson you will connect the SNAP Ultimate I/O unit to an OPC client. OPC stands for OLE (Object Linking and Embedding) for Process Control and is an open format developed by the OPC Foundation (of which Opto 22 is one of six founding members). The OPC foundation creates the standards enabling manufacturers of software and hardware products to create inter-compatible products for industrial use. For example, using OPC standards, you can build a system of I/O; controllers; and monitoring, alarming, and trending software all from different manufacturers. This lesson shows how to set up the OptoOPCServer to allow SNAP Ultimate I/O to be used with any OPC-compatible software. In this scenario, the utility OPC Test Client acts as the OPC client software.



CONCEPTS

OptoEnetServer and OptoOPCServer

Opto 22 has two related server applications:

- **OptoEnetServer** is an OPC 1.0-compliant server that lets clients using OPC or DDE (Dynamic Data Exchange) to communicate with one or more SNAP Ethernet-based brains. Use OptoEnetServer if you are using DDE. If you are using OPC, use OptoOPCServer instead.
- **OptoOPCServer** is an OPC 2.0 compliant server that provides OPC clients access to Opto 22 SNAP Ethernet-based brains and allows client applications to read and

write to I/O points or even change their specific configuration options. **Note: This version is an OPC server only and does not include DDE functionality.**

How OptoOPCServer works

OptoOPCServer acts as an OPC translator by:

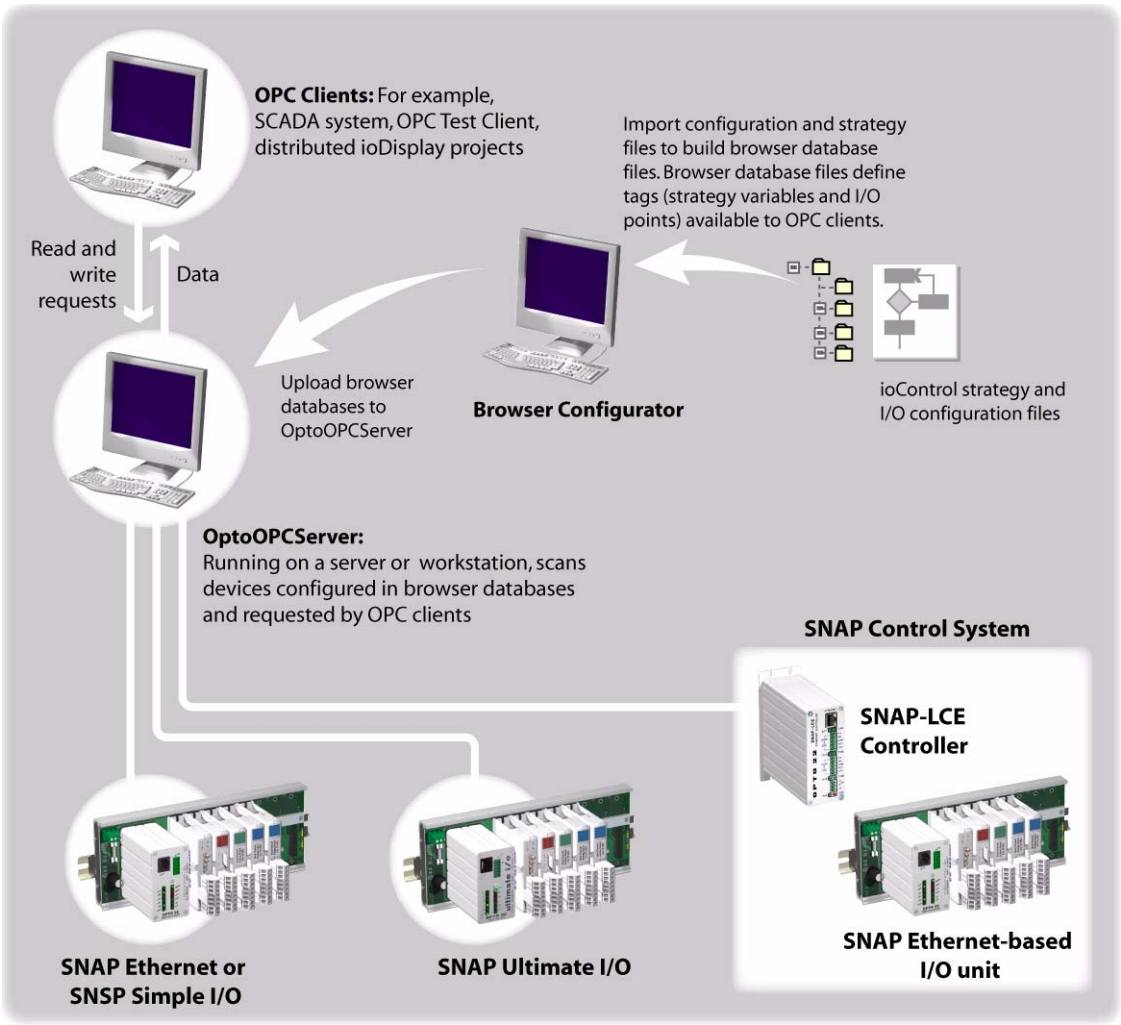
- Translating incoming OPC messages into the memory-mapped format used by the I/O side of SNAP Ethernet-based brains, including those in the SNAP Ultimate I/O, SNAP Ethernet I/O and SNAP Simple I/O product families.
- Translating incoming OPC messages into the form used by SNAP Ethernet-based industrial controllers, such as the control engine side of the SNAP Ultimate brain, the SNAP-LCM4 controller (with M4SENET-100 Ethernet adapter card), or the SNAP-LCE control engine.
- Translating outgoing messages from SNAP Ethernet-based controllers and brains into the OPC format.

OptoOPCServer can be used to integrate SNAP Ethernet-based controllers and brains with PCs running Microsoft products, OPC-compliant third-party packages, and custom applications you create with tools such as Visual C++™. Since OptoOPCServer can communicate with both I/O units and controllers, it provides data from both I/O points and strategy variables.

The server can run on the same computer as the OPC client or on another computer within the same Ethernet network. Any OPC client on the network can attach to the server and communicate with the controllers or brains and the I/O points they control and monitor.

OptoOPCServer continually scans and stores information from the controllers, brains, and I/O, providing real-time information to multiple PCs needing data from the same processor, without slowing down the processors throughput.

OptoOPCServer System Architecture



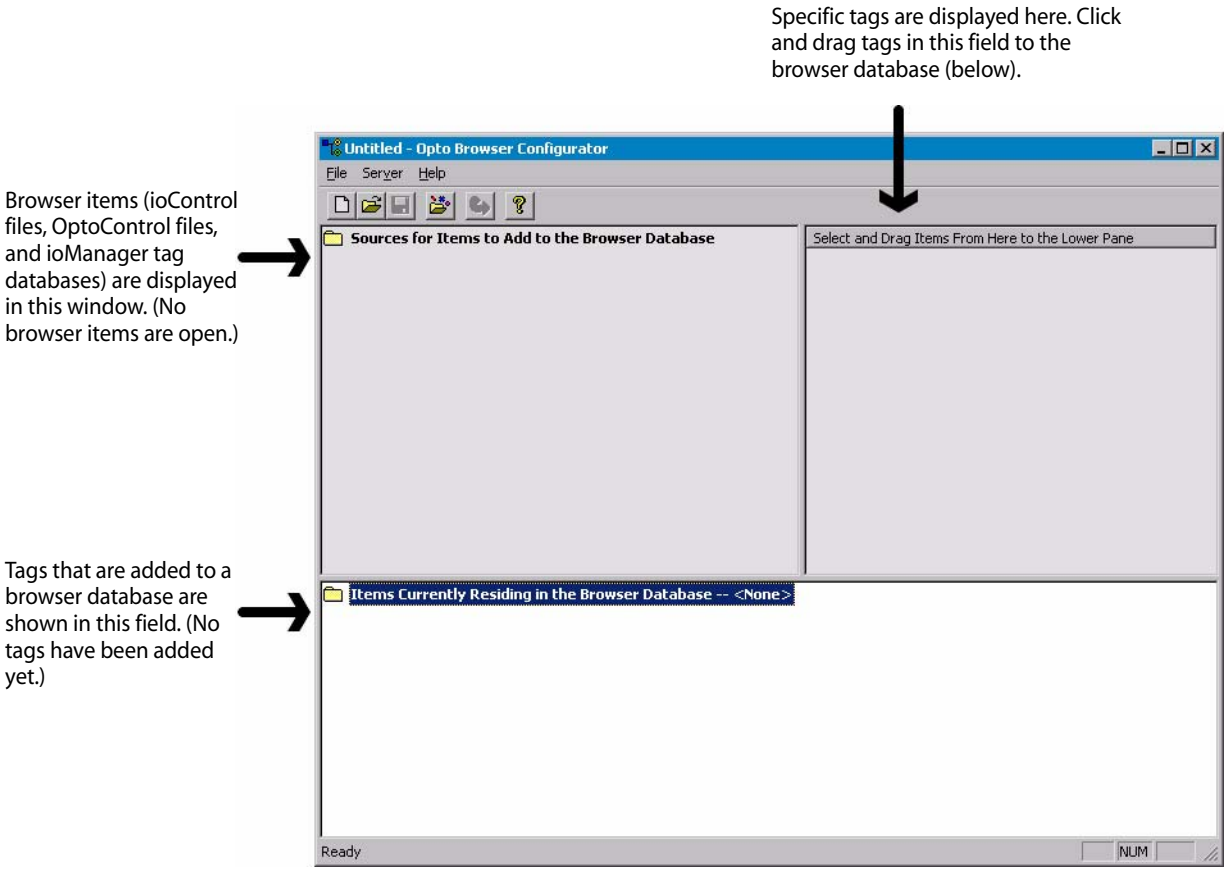
ACTIVITY

In this activity, you use the Browser Configurator to create a browser database of tags that the third-party OPC Test Client (included with OptoOPCServer) can read and write to by way of the OptoOPCServer. While the procedure for defining tags will vary with the OPC client, using the test client is useful for verifying that the OptoOPCServer is working properly.

Creating a Browser Database

1. Launch Opto Browser Configurator.

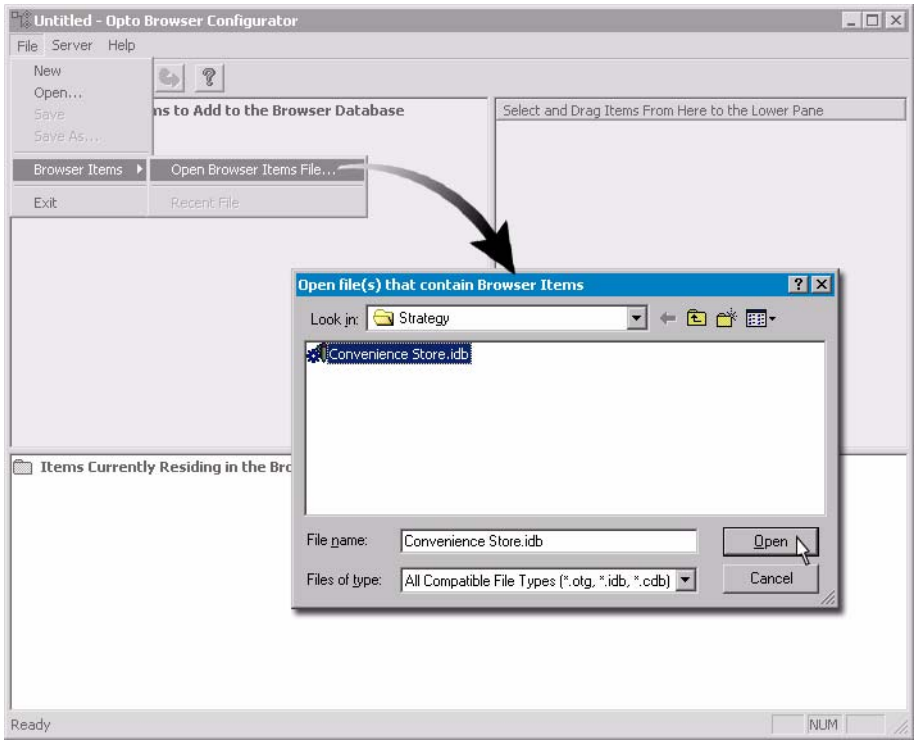
Choose *Start* ⟶ *Programs* ⟶ *Opto 22* ⟶ *ioProject Software* ⟶ *OptoOPCServer* ⟶ *Browser Configurator.*



2. Select the ioControl strategy file.

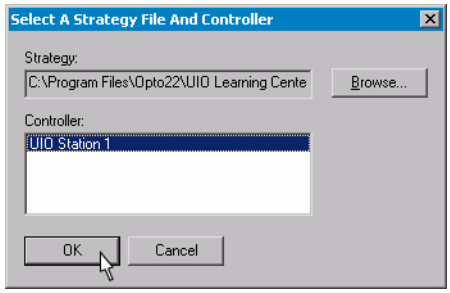
a. Select *File* ⟶ *Browser Items* ⟶ *Open Browser Items File*.

b. In the Open files(s) dialog box, navigate to the **C:\Program Files\Opto22\UIO Learning Center\Strategy** folder.



c. Select **Convenience Store.idb** and click *Open*.

d. In the Select a Strategy File And Controller dialog box, confirm that the correct strategy and control engine are selected, and click *OK*.



The Opto Browser Configurator window

The I/O unit on the SNAP Ultimate controller/brain contains values, states, configuration, and features such as the latches of a digital point shown here.

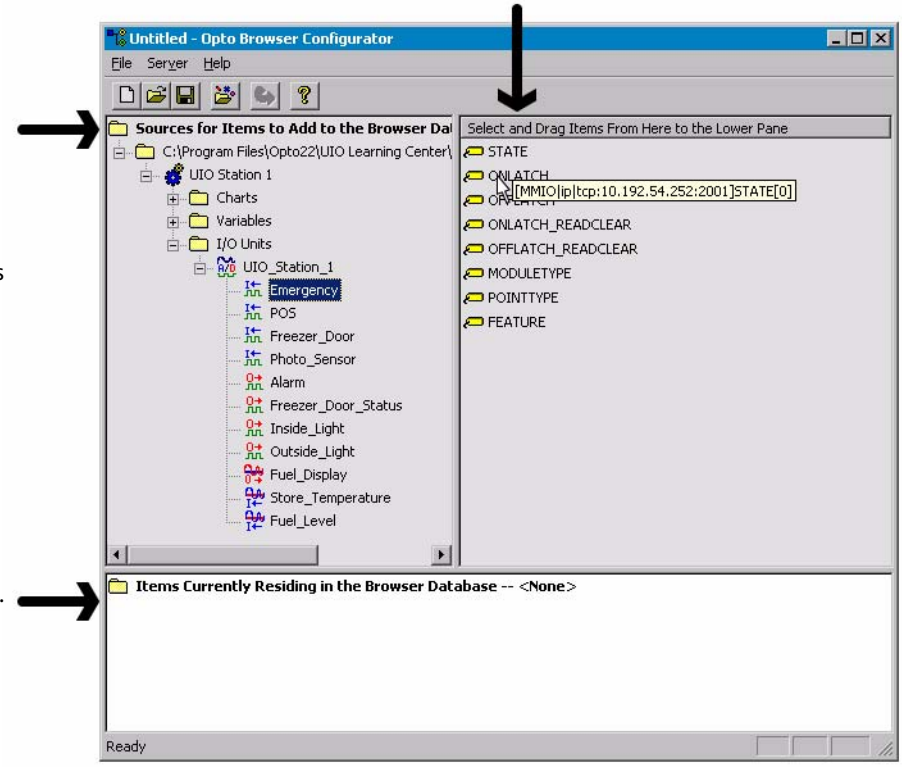
The OptoOPCServer provides access to all of this information. All of the points shown here have similar features.

Tags associated with the item selected in the left pane. To see the OPC item ID for an item, move the mouse over the item. (For more information on item IDs, see the *OptoOPC Server User's Guide*.)

Browser Items are shown here. Browser items are ioControl strategies and I/O configuration files that define the tags available from the Opto 22 hardware.

Clicking the plus and minus signs next to folders expands or collapses the contents. Clicking on a folder or item displays its tags in the right-hand pane.

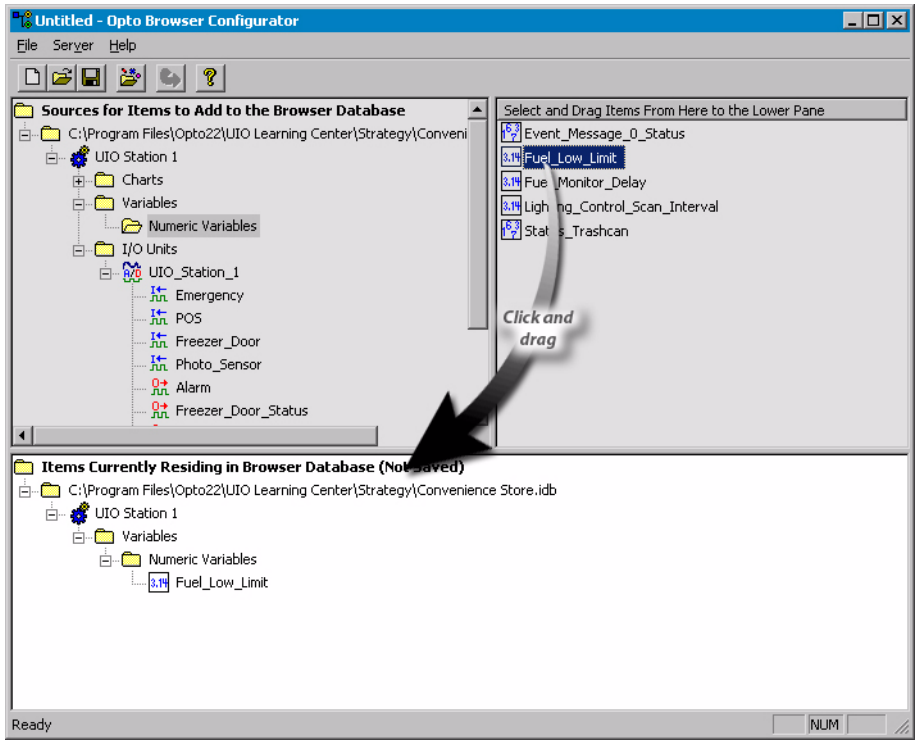
The contents of the browser database file are shown here. Drag tags into this window to build a browser database.



3. Add a control engine tag to the browser database.

a. Expand the *Variables* folder.
b. Click the *Numeric Variables* folder.

c. Drag the *Fuel_Low_Limit* variable from the right-hand pane and drop it into the lower section labeled "Items Currently Residing in the Browser Database."

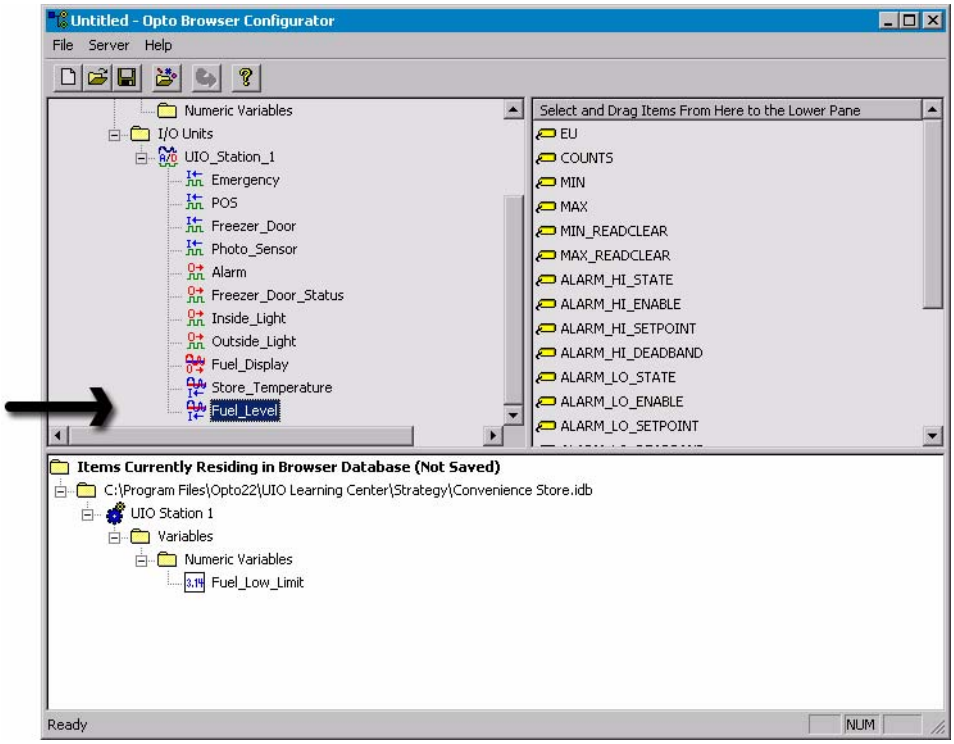


4. Add an I/O tag to the browser database.

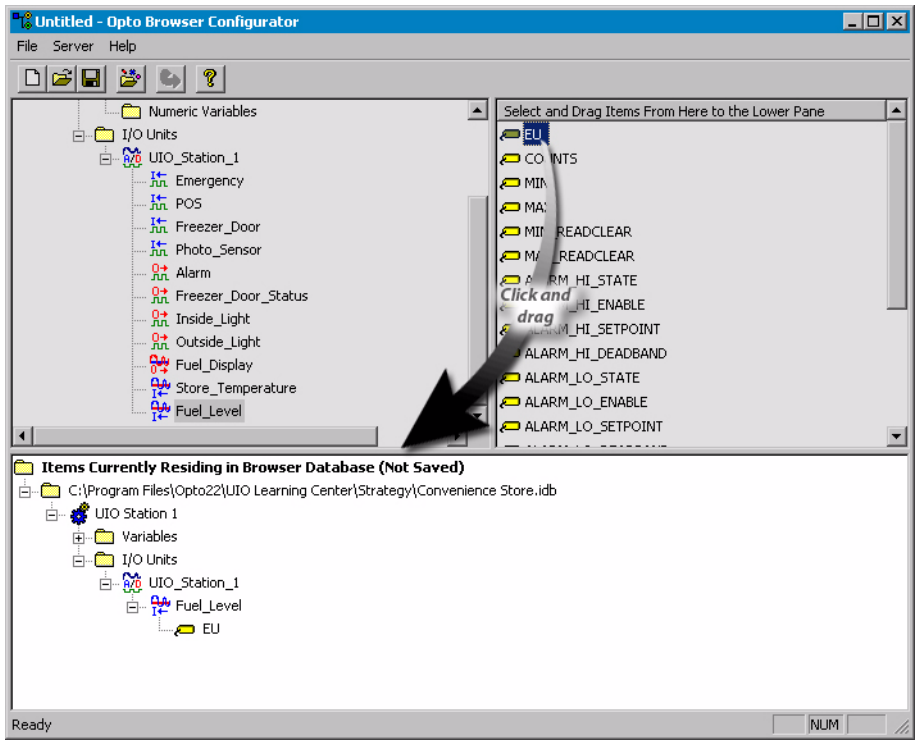
a. Expand the *I/O Units* folder.
b. Expand *UIO_Station_1*.

c. Select *Fuel_Level*. (You may need to scroll.)

Select Fuel_Level here to display all the tags associated with this I/O point.

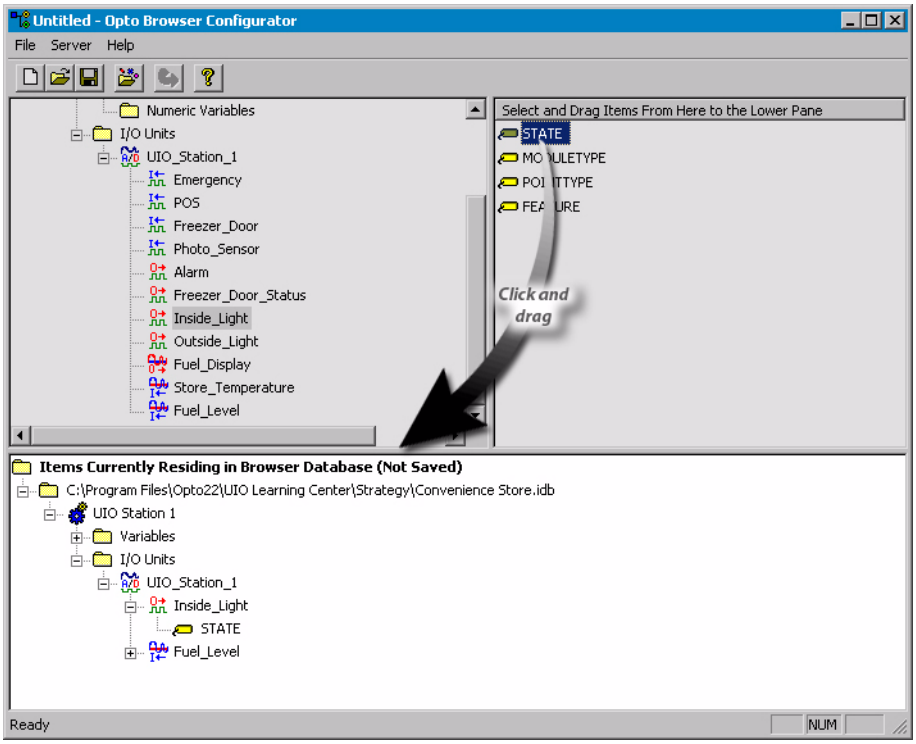


d. In the upper right-hand pane, select *EU* (Engineering Units) and drag it down to the lower pane. Fuel_Level is now added to the Browser Database.



e. In the I/O Units folder in the upper left-hand pane, select *Inside_Light*.

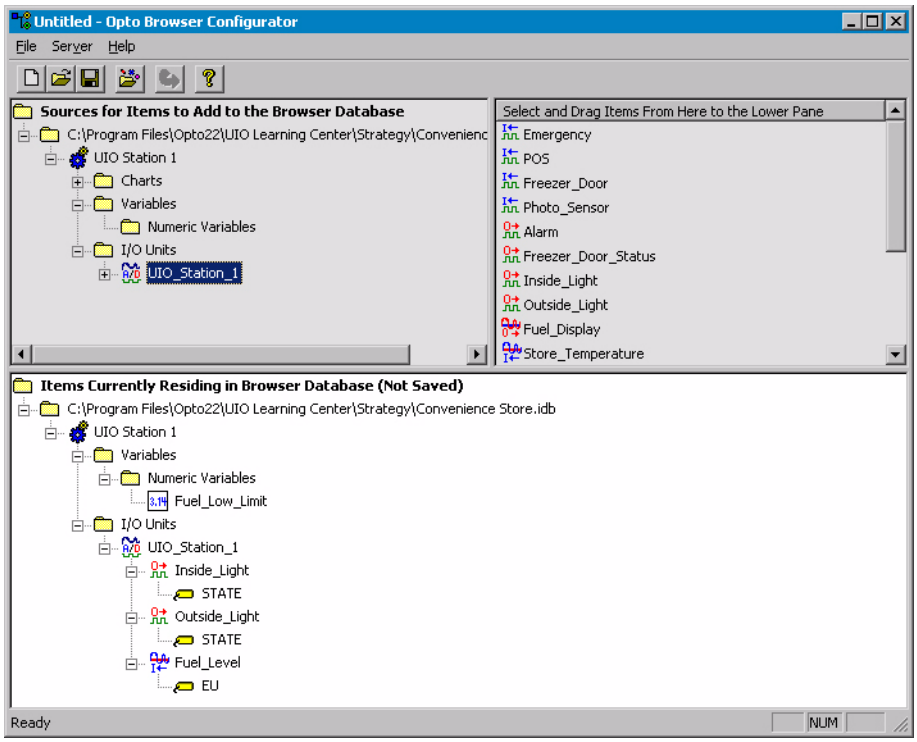
f. In the upper right-hand pane, select *State* and drag it down to the lower pane.



Inside_Light is now added to the Browser Database.

5. Add Outside_Light to the browser database.

a. Repeat the previous step for Outside_Light. You should now have four items in your Browser Database.



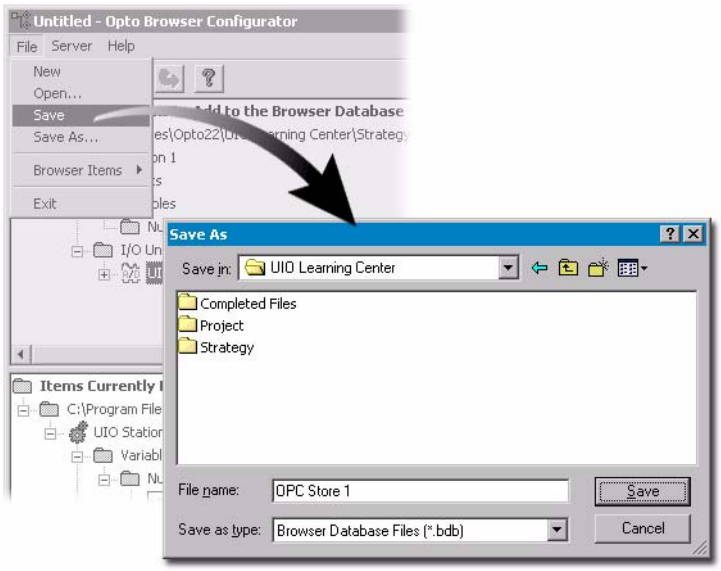
b. Feel free to add more items. If you need to delete an item, click it in the "Items Currently Residing in Browser Database" field and press the delete key on your keyboard.

6. Save the browser database.

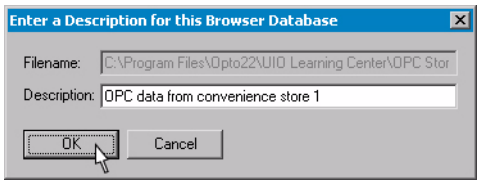
a. To save the configuration, select *File* ⟶ *Save*.

b. Navigate to the **C:\Program Files\Opto22\UIO Learning Center** folder.

c. Type `OPC Store 1` as the File name, and click *Save*.

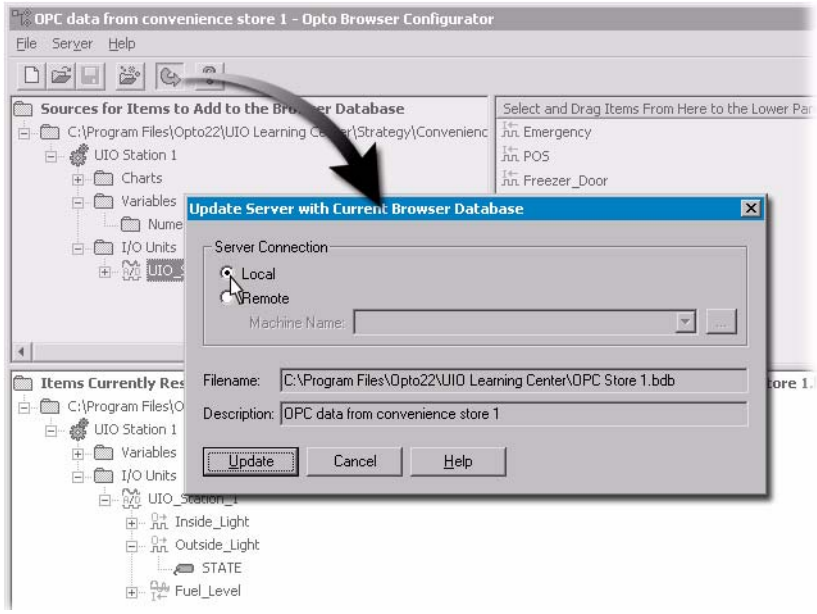


d. Type a description, and click *OK*.

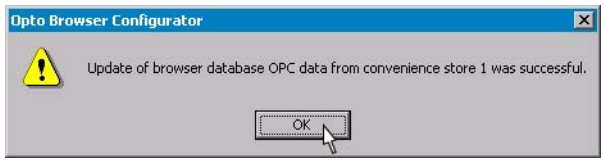


e. On the toolbar, click the *Update Server* button ( ) to publish the current browser configuration to the OPC server.

Because you will be connecting to OptoOPCServer running on your own PC, you will use the default Local connection.



f. Click *Update*.
g. Click *OK*.



7. Close the Opto Browser Configurator.

a. Select *File* ➞ *Exit*.

> Normally you can exit the server, since launching the client automatically launches the server, too.
>
> If you want to see communication in the server window for debugging purposes, however, do not exit the server. Make sure it is open before launching the client.

Configuring the OPC Test Client

1. Launch the OptoOPC Test Client.

Select *Start* ➞ *Programs* ➞ *Opto 22* ➞ *ioProject Software* ➞ *OptoOPCServer* ➞ *OPC Test Client*.

2. Connect to OptoOPCServer.

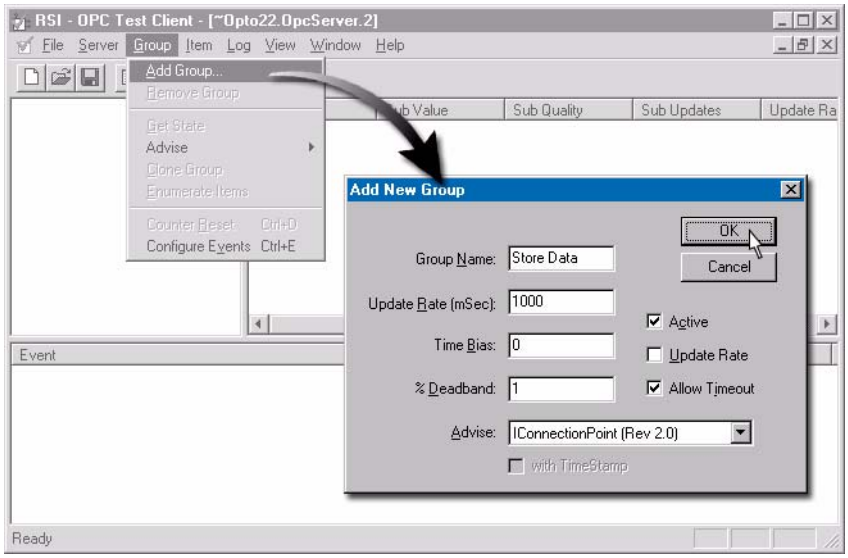
a. Select *Server* ➞ *Connect*.
b. Select **Opto22.OpcServer.2** (company name, product name, version). Since the server and client are running on the same computer, you can leave the Node Name field blank.

c. Click *OK* to close the Select an OPC Server dialog box.

3. Add a group.

The client requires that you designate a group for the items the client is requesting. Groups provide a way to separate items by the rate at which they need to be scanned. For example, temperature in a small chamber may need to be scanned very frequently, while an outdoor temperature may change much more slowly and need less frequent scanning. These two tags would be in different groups. You can add as many groups as you need.

a. From the Group menu, select *Add Group*.

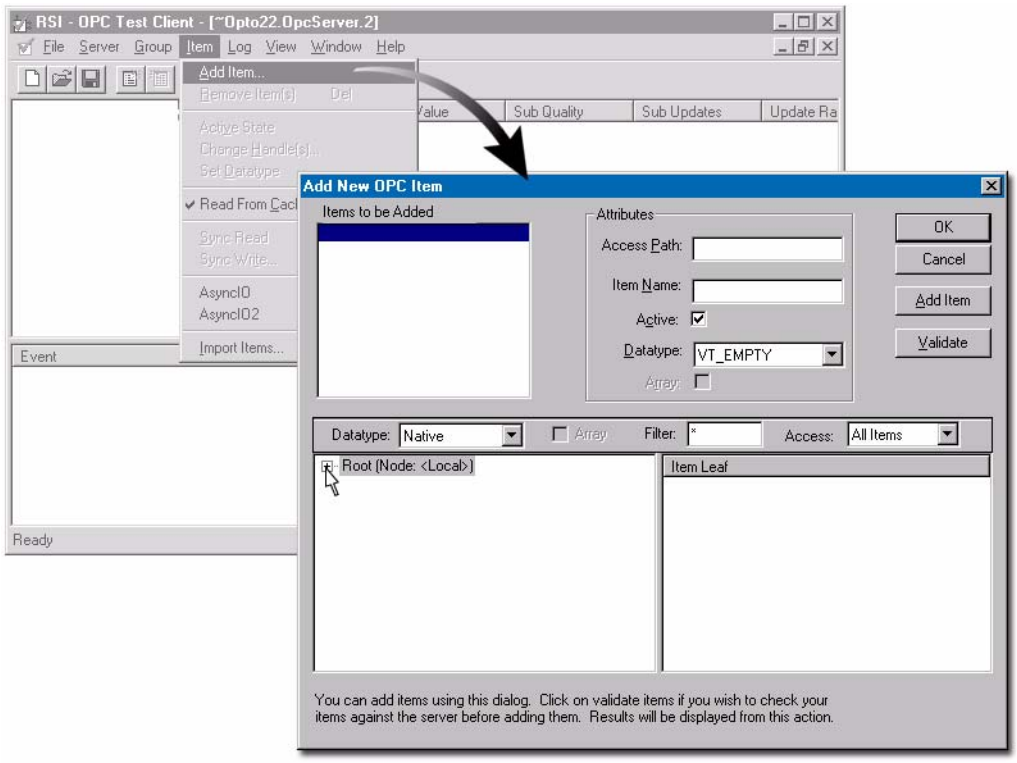


b. Type `Store Data` in the Group Name field.
c. Type `1000` in the Update Rate (mSec) field. OptoOPCServer will scan this group every second.
d. Click *OK*.

The new group appears in the main window.

4. Add the Freezer Door tag to the group.

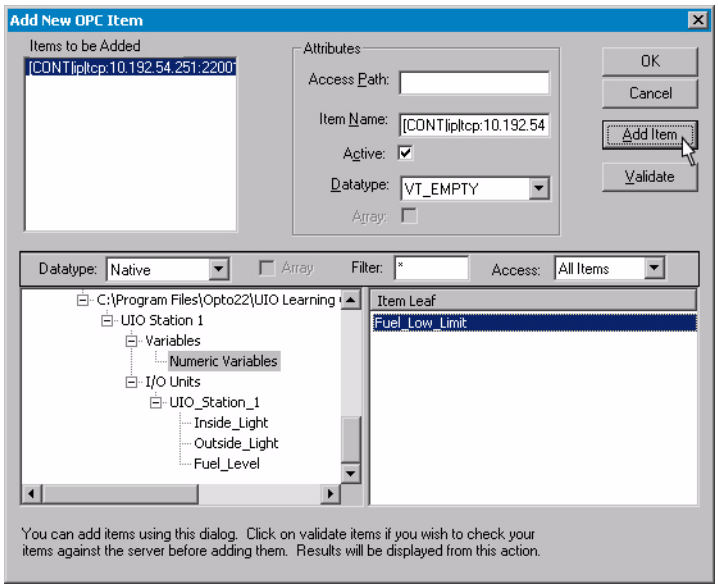
a. Select *Item* ⟶ *Add Item*.



b. In the lower left part of the window, expand the Root Node folder until you can select the Numeric Variables folder.

c. Select the Numeric Variables folder and click *Fuel_Low_Limit*.

The ITEM ID appears in the Item Name field and in the Items to be Added list.

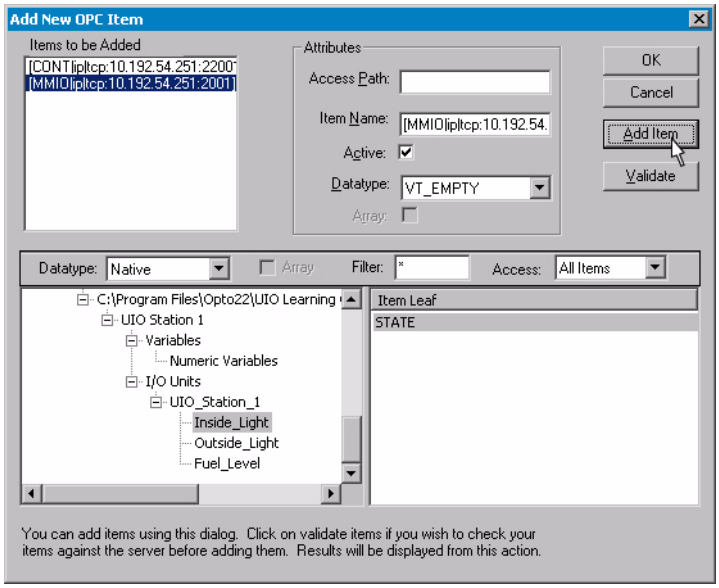


d. Click *Add Item*.

e. Leave the Add New OPC Item dialog box open.

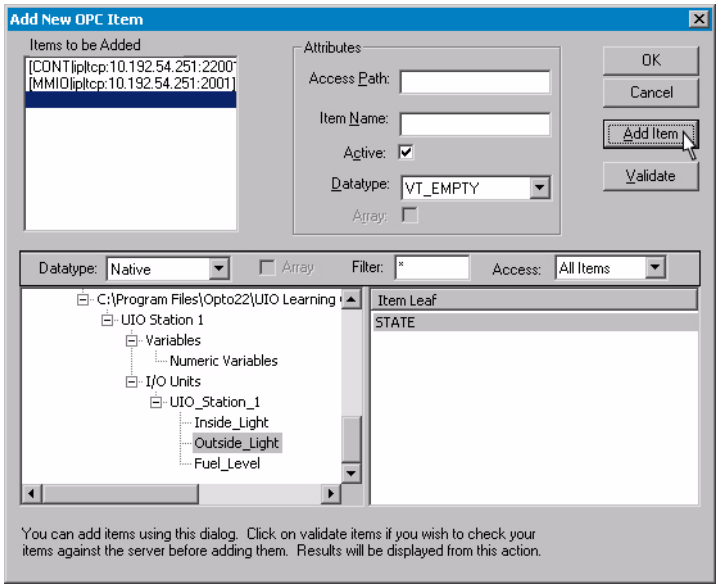
5. Add Inside_Light (State).

a. Expand the I/O Units folder.
b. Expand the UIO_Station_1 folder.
c. Click *Inside_Light*.
d. Click *State* in the Item Leaf list.
e. Click *Add Item*.



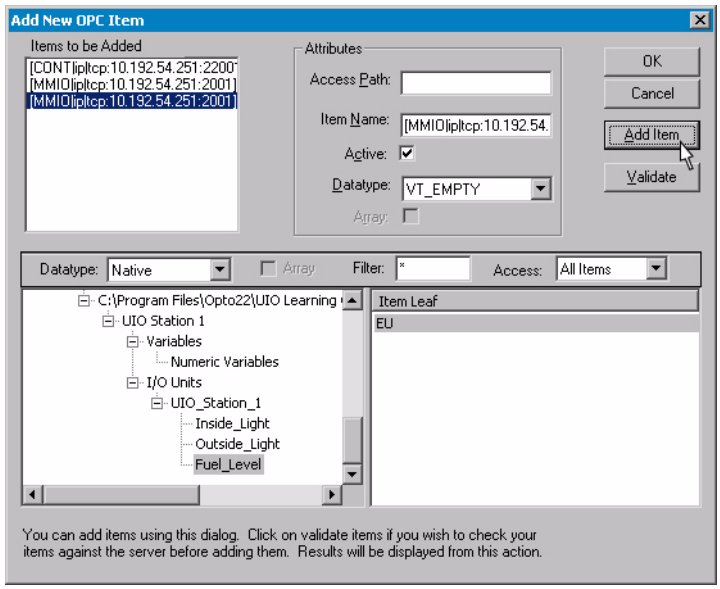
6. Add Outside_Light (State).

a. Click *Outside_Light*.
b. Click *State* in the Item Leaf list.
c. Click *Add Item*.



7. Add Fuel_Level (EU).

a. Click *Fuel_Level*.
b. Click *EU* in the Item Leaf list (EU = Engineering Units).
c. Click *Add Item*.



8. Close the Add New OPC Item dialog box.

Click *OK* when you are done adding items.

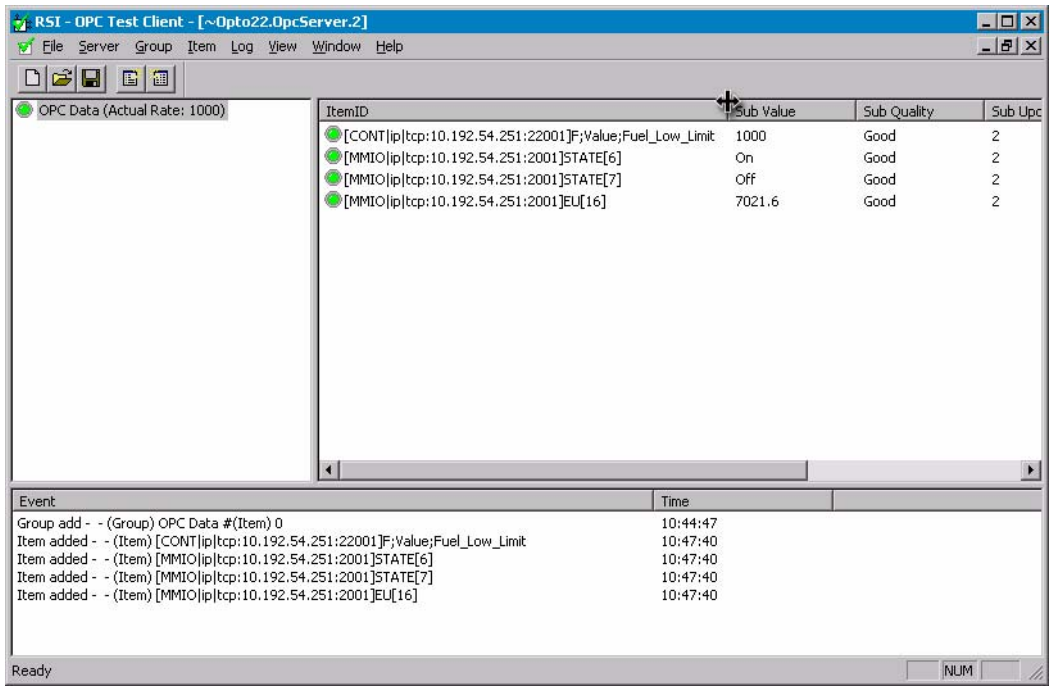
Testing Your Work

The main client window shows the items you have added. If the connection is successfully completed, you can watch the time values being updated in the Sub Value column. If the server cannot connect to the IP address, the Sub Quality column shows "Bad Communication" for that item. You can add more groups and items at any time.

1. Read tags.

a. Resize the Item ID column by dragging the column marker to the right.

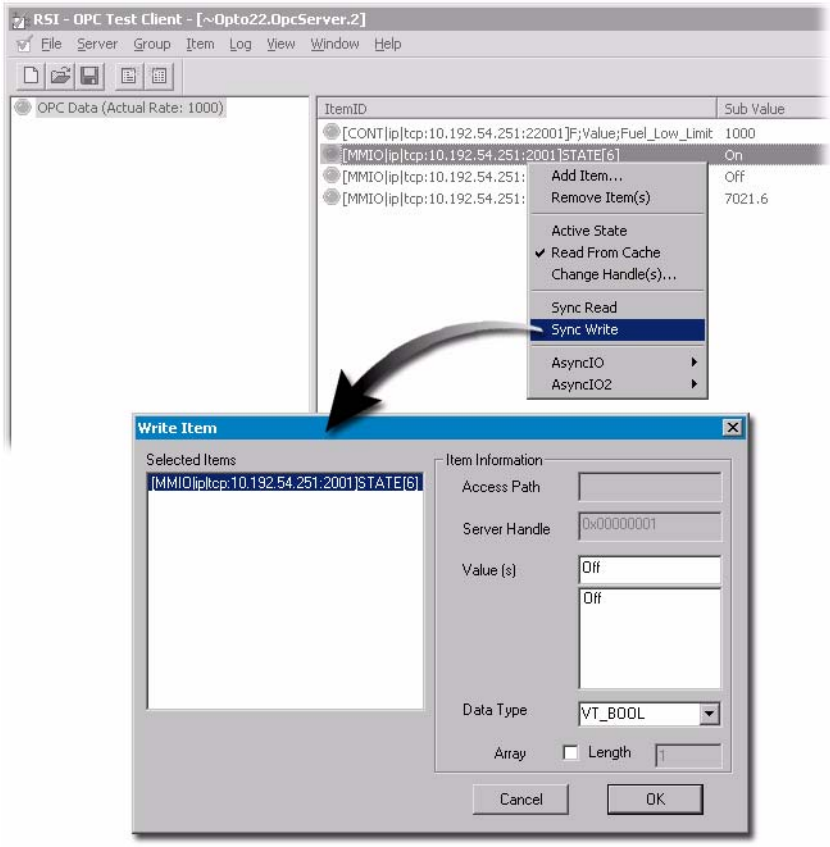
b. Turn the potentiometer for the Fuel_Level and watch the sub value change.



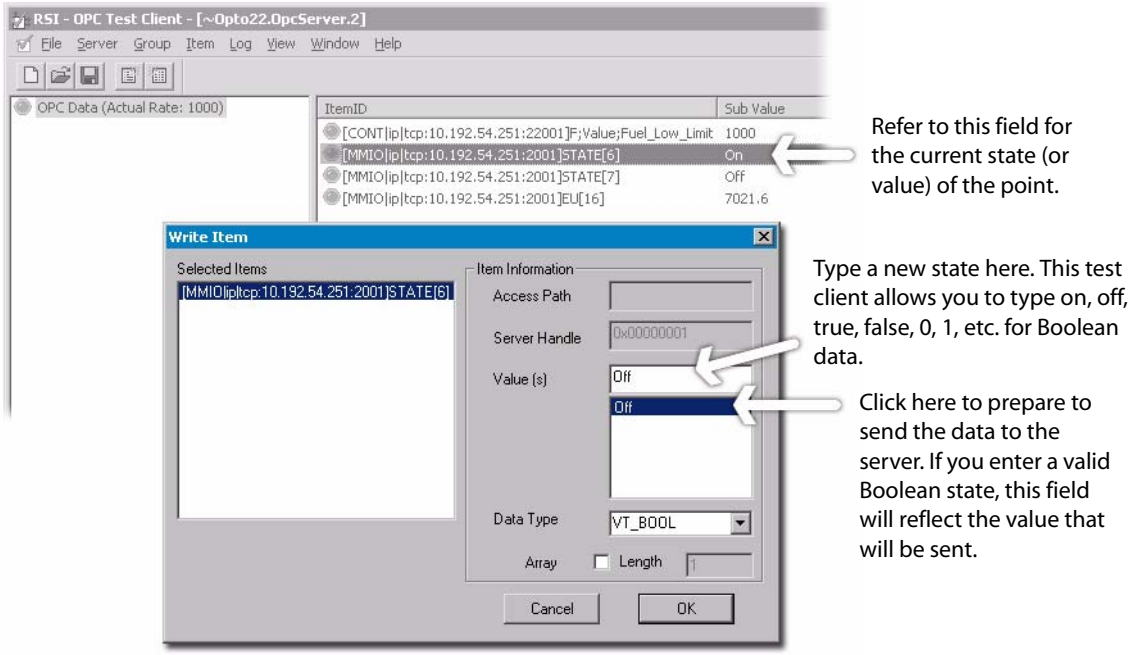
2. Write to a digital point.

a. Right-click the second item (State[6]).

b. Select *Sync Write*.



c. In the Write Item dialog box, type the new state in the Value(s) field: If your point is on, type `Off`; if it is off, type `On`.



d. Click the field just below the Values field.

e. Click *OK*.

Observe the indoor light change state.

f. Repeat the procedure, if desired.

> **Synchronous and Asynchronous**
>
> For output points, you can do a synchronous write or an asynchronous write. In synchronous writes, the client waits for a reply from the server. In asynchronous writes, the client doesn't wait for a reply. Which method you would use depends on the type of write and the typical communication time between your client application, the OptoOPCServer, and the SNAP Ultimate I/O unit.

3. Save your file.

a. Click *File* ➞ *Save*.
b. Navigate to the **C:\Program Files\Opto22\UIO Learning Center** folder.
c. Type `ConvStore.csv` for the file name.
d. Click *Save*.

4. Close the Test Client.

Debugging in the OptoOPCServer Window

You may find data in the OptoOPCServer window useful for debugging. The server is normally launched by the client and runs invisibly; in order to see its data, you must start it BEFORE starting the client. You cannot open the server window after the client has launched the server. For more information, see *OptoOPCServer User's Guide*.

1. Close OptoOPCServer, Opto Browser Configurator, and OptoOPCServer, if any of them are running.

2. Open OptoOPCServer.

Start ➞ *Programs* ➞ *Opto 22* ➞ *ioProject Software* ➞ *OptoOPCServer* ➞ *Opto OPCServer*.

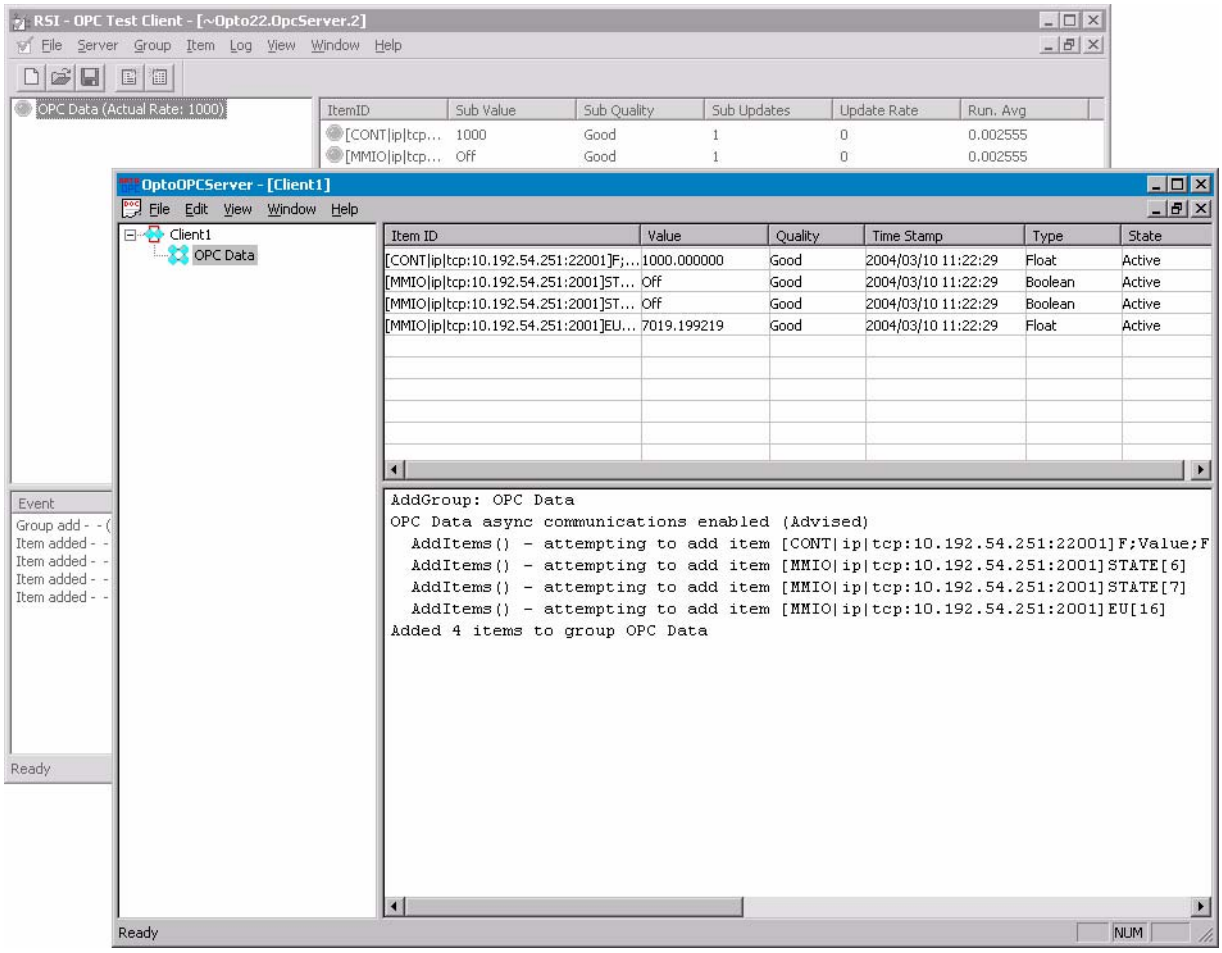
3. Open the OPC Test Client.

Start ➞ *Programs* ➞ *Opto 22* ➞ *ioProject Software* ➞ *OptoOPCServer* ➞ *OPC Test Client*.

4. Open your test client file.

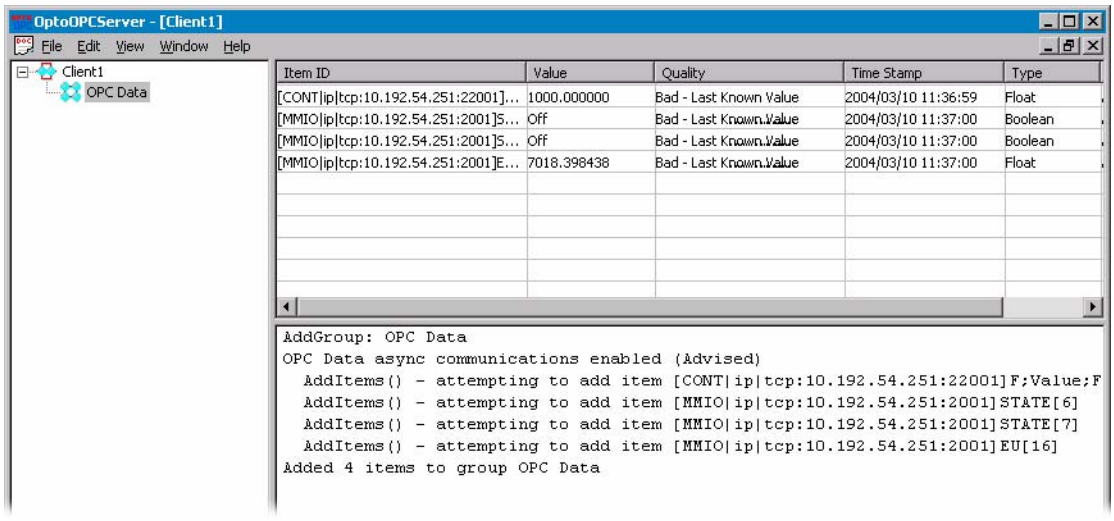
a. Click *File* ➞ *Open*.
b. Navigate to the **C:\Program Files\Opto22\UIO Learning Center** folder.
c. Select `ConvStore.csv`.
d. Click *Open*.

5. Switch to OptoOPCServer.



Connection messages and live data updates confirm that OptoOPCServer is getting data from the SNAP Ultimate I/O unit.

6. **Turn off your Learning Center.**



The Quality and Time Stamp fields report that the data represent the last known values.

7. **Turn on your Learning Center.**

After a few seconds, OptoOPCServer reconnects.

8. **Close all applications.**

CONCLUSION

Your introduction to the main applications of the ioProject software suite is complete. Congratulations for your dedication to this tutorial. In the next chapter, you'll find a roadmap of the concepts and features you applied in this tutorial. This roadmap may be helpful should you wish to return to the samples you built and review the material.

ROAD MAP

The following pages present a graphical overview of the ioControl strategy and ioDisplay project developed in the SNAP Ultimate I/O Learning Center guide.

All components of the strategy are listed on the left side. The chapters of this guide are represented as columns. Follow a horizontal path to the right of each component to see where in the lessons the component is introduced, used, and modified. This overview also shows which components affect subsequent chapters (for example, which ioControl components are critical to the ioDisplay project).

<Lesson/Chapter:>

<intro> Getting Started

<1> ioManager

<2> Introduction to ioControl

<3> Flowcharts

<4> Digital Control

Opto 22 Software

ioManager

Store IO Config.otg

ioControl

Convenience Store.idb

ioDisplay

Monitor Convenience Store

OptoOPCServer

Concepts:
- Assembling SNAP Systems
- Networking SNAP Systems

ioManager

Concepts:
- IP addressing
- Firmware

Control Engine

UIO Station 1

Charts

Powerup

Lighting_Control

Fuel_Monitor

Variables

Event_Message_0_Status
Fuel_Monitor_Delay
Fuel_Low_Limit
Lighting_Control_Scan_Interval
Status_Trashcan

I/O Units

UIO_Station_1

	Points
0	00 Emergency
	01 POS
	02 Freezer_Door
	03 Photo_Sensor
1	04 Alarm
	05 Freezer_Door_Status
	06 Inside_Light
	07 Outside_Light
2	08 Fuel_Display
3	12 Store_Temperature
4	16 Fuel_Level

EVENTMESSAGE,0,
Regular Fuel Low

E-Mail

File Management
XML

ioControl | Flowcharts | Digital

Convenience Store.idb

Concepts:
- Configuring a controller
- Configure mode

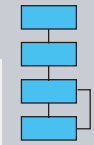
Concepts:
- Drawing charts
- Debug mode
- Forcing point states
- Detaching points

Configure control engine:
UIO Station 1

Powerup

Start chart: Lighting Control

Lighting_Control
Start building chart



Finish chart

Concepts:
- Loopback address
- Downloading a strategy
- Autorun
- Configuring I/O

Status_Trashcan

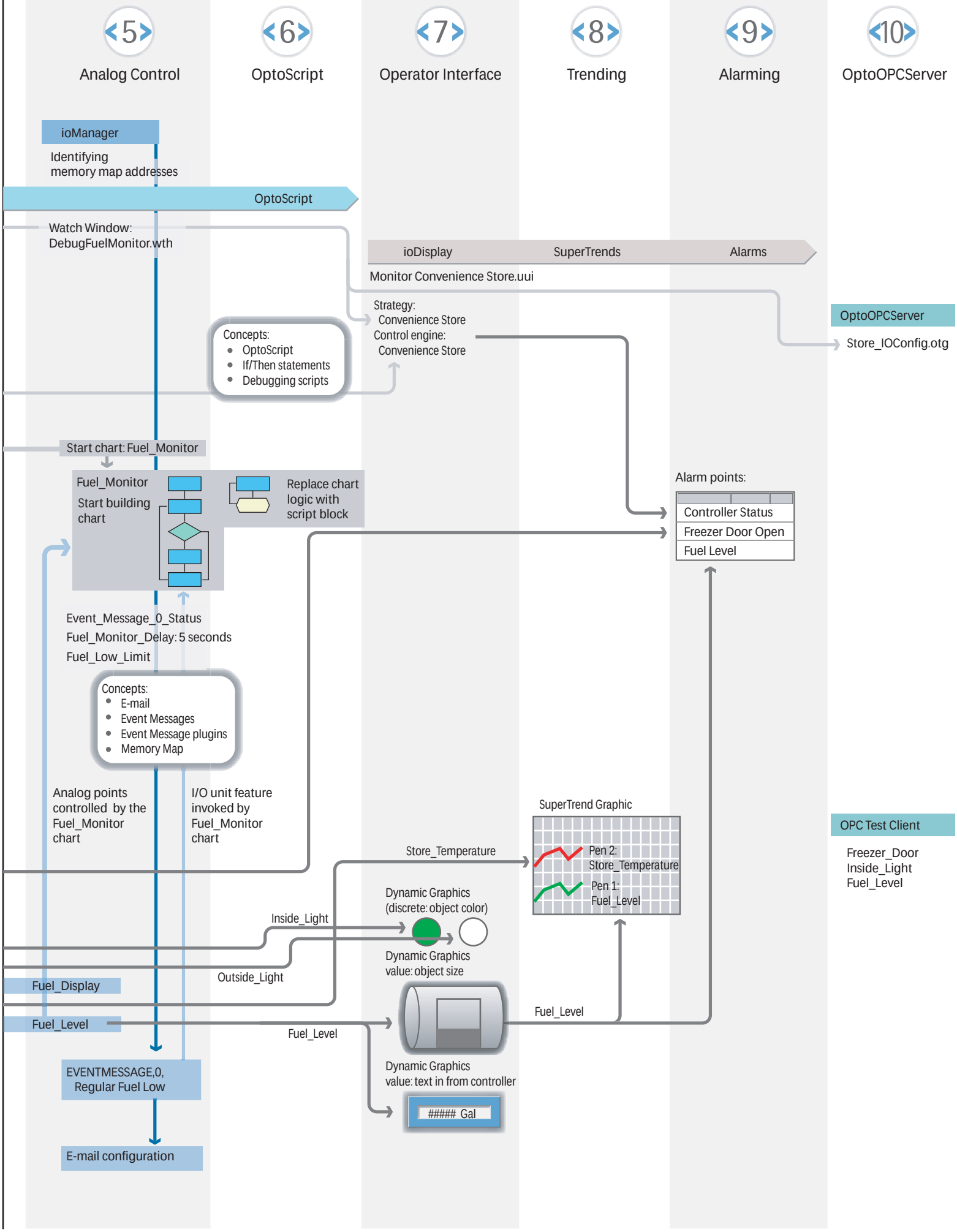
Lighting_Control_Scan_Interval
2 seconds

Digital points controlled by the Lighting_Control chart

Configure I/O Unit:

UIO_Station_1

Points		
00 Emergency		Emergency
01 POS		
02 Freezer_Door		Freezer_Door
03 Photo_Sensor		Photo_Sensor
04 Alarm		Alarm
05 Freezer_Door_Status		Freezer_Door_Status
06 Inside_Light		Inside_Light
07 Outside_Light		Outside_Light
08 Fuel_Display	Scale: -10000/10000	
12 Store_Temperature		
16 Fuel_Level	Scale: -20000/20000	


5
Analog Control
6
OptoScript
7
Operator Interface
8
Trending
9
Alarming
10
OptoOPCServer
ioManager
Identifying
memory map addresses
OptoScript
Watch Window:
DebugFuelMonitor.wth
ioDisplay
SuperTrends
Alarms
Monitor Convenience Store.uui
Strategy:
Convenience Store
Control engine:
Convenience Store
OptoOPCServer
Store_IOConfig.otg
Concepts:
OptoScript
If/Then statements
Debugging scripts
Start chart: Fuel_Monitor
Fuel_Monitor
Start building
chart
Replace chart
logic with
script block
Alarm points:
Controller Status
Freezer Door Open
Fuel Level
Event_Message_0_Status
Fuel_Monitor_Delay: 5 seconds
Fuel_Low_Limit
Concepts:
E-mail
Event Messages
Event Message plugins
Memory Map
Analog points
controlled by the
Fuel_Monitor
chart
I/O unit feature
invoked by
Fuel_Monitor
chart
SuperTrend Graphic
Pen 2:
Store_Temperature
Pen 1:
Fuel_Level
OPC Test Client
Freezer_Door
Inside_Light
Fuel_Level
Store_Temperature
Dynamic Graphics
(discrete: object color)
Inside_Light
Outside_Light
Dynamic Graphics
value: object size
Fuel_Display
Fuel_Level
Fuel_Level
Fuel_Level
EVENTMESSAGE,0,
Regular Fuel Low
Dynamic Graphics
value: text in from controller
Gal
E-mail configuration