**NATIONAL INSTRUMENTS™**

**LabVIEW™**

# LabVIEW Datalogging and Supervisory Control Module Developer's Manual

**NATIONAL INSTRUMENTS™**

**Worldwide Technical Support and Product Information**

`ni.com`

**National Instruments Corporate Headquarters**

11500 North Mopac Expressway    Austin, Texas 78759-3504    USA    Tel: 512 683 0100

**Worldwide Offices**

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China (Shanghai) 021 6555 7838,
China (ShenZhen) 0755 3904939, Czech Republic 02 2423 5774, Denmark 45 76 26 00,
Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30, Greece 30 1 42 96 427,
Hong Kong 2645 3186, India 91805275406, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970,
Korea 02 596 7456, Malaysia 603 9596711, Mexico 001 800 010 0793, Netherlands 0348 433466,
New Zealand 09 914 0488, Norway 32 27 73 00, Poland 0 22 528 94 06, Portugal 351 1 726 9011,
Russia 095 2387139, Singapore 2265886, Slovenia 386 3 425 4200, South Africa 11 805 8197,
Spain 91 640 0085, Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2528 7227,
United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the
documentation, send e-mail to `techpubs@ni.com`.

# Important Information

# Conventions

This manual uses the following conventions:

» The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a tip, which alerts you to advisory information.

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

**bold** Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

*italic* Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

`monospace italic` Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

# Contents

# Chapter 4
# Using Tags to Manage I/O in LabVIEW

# Chapter 5
# Displaying and Controlling Data

# Chapter 6
# Alarms and Events

# Chapter 7
# Historical Data Logging and Extraction

# Chapter 8
# Security

# Chapter 9
# Networking and Deploying Applications

# Appendix A
# Using SQL to Access Historical Data in Citadel

# Appendix B
# Technical Support Resources

# Glossary

# Index

# 1

# Introduction

The LabVIEW Datalogging and Supervisory Control (DSC) module adds features and capabilities to LabVIEW to help you create automation applications or applications that must keep track of a large number of data points. The LabVIEW DSC module provides solutions for supervisory control of a wide variety of distributed systems using the flexibility of graphical LabVIEW programming. Before you use the LabVIEW DSC module, you should be familiar with LabVIEW.

Use the LabVIEW DSC module to change setpoints or send control instructions to individual devices while monitoring the entire system. The LabVIEW DSC module provides the following features and capabilities:

- Configuration utilities and wizards
- Real-time database
- Historical data collection and trending
- Alarm and event reporting and logging
- Security
- Connection to PLC and industrial device networks
- OPC server and client
- Connection to a wide selection of device servers

## Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- LabVIEW documentation, including the *LabVIEW Help.* Select **Help»Contents and Index** to access the *LabVIEW Help* and help files you create. Refer to the *LabVIEW Help* for more information about creating your own help files.

- *LabVIEW Datalogging and Supervisory Control Module Help.* Select **Help»Datalogging and Supervisory Control** to access more information about the LabVIEW DSC module palettes, tools, VIs, and

functions. The *LabVIEW Datalogging and Supervisory Control Module Help* also includes step-by-step instructions for using the LabVIEW DSC module features.

•   *Getting Started with the LabVIEW Datalogging and Supervisory Control Module*. This manual gives an introduction to the LabVIEW DSC module and includes a brief tutorial.

# Controls Palette

The LabVIEW DSC module installs the following subpalettes on the **Controls** palette, as shown in Figure 1-1:

•   **Vessels**—Several tanks, a hopper, and a bin. All objects on this subpalette are numeric.

•   **Pipes, Pumps, & Valves**—Various Boolean pipe, pump, and valve objects.

•   **Alarms and Events**—A set of display, format, query, and summary tools for handling alarms and events.

•   **Trends**—Historical trend XY graph, real-time trend waveform chart, and NI HyperTrend control.

•   **DSC Server Data Types**—Various data type definitions you can use to write VI-based servers for the LabVIEW DSC module. Refer to the *Developing Device Servers* section of Chapter 3, *Servers*, for more information about developing device servers.

| 1 | Vessels | 4 | Trends |
| 2 | Pipes, Pumps, and Valves | 5 | DSC Server Data Types |
| 3 | Alarms and Events | | |
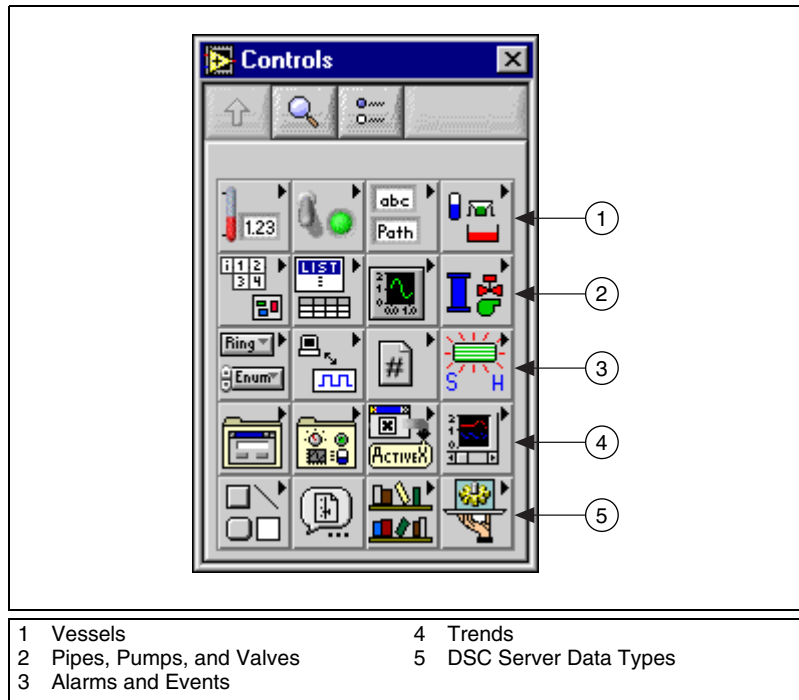
**Figure 1-1.** Controls Palette with the LabVIEW DSC Module

The LabVIEW DSC module also installs the tag control on the
**Controls»I/O** palette. Use the tag control to select a tag from those
configured in the active configuration (.scf) file.

# Functions Palette

The LabVIEW DSC module installs the following subpalettes on the
**Functions** palette, as shown in Figure 1-2. Refer to the *LabVIEW
Datalogging and Supervisory Control Module Help* for more information
about the VIs and functions on these subpalettes.

- **Tags**—Use these VIs and functions to read the most recent value of a
  tag, write a new value to a tag, or obtain data for a real-time trend.

- **Alarms & Events**—Use these VIs to acknowledge alarms, display
  alarm summary or event history information, query logged alarm and
  event data, or obtain alarm summary status.

- **Tag Attributes**—Use these VIs to read and write tag configuration
  parameters programmatically.

- **Historical Data**—Use these VIs to read and write historical data, resample trend data, compute statistical data for a historical trend, convert historical trend data to a spreadsheet, or perform some other database administrative task.

- **Historical Data»Data Set Logger**—Use these VIs to retrieve information or values for sets of logged data.

- **System**—Use these VIs to launch or shut down the Tag Engine, to enable or disable event logging, historical datalogging, or printing, or to manipulate the LabVIEW DSC module environment.

- **System»Security**—Use these VIs to access security information about your application.

- **DSC Server Development**—Use these VIs to develop VI-based device servers. Refer to the *Developing Device Servers* section of Chapter 3, *Servers*, for more information about developing device servers.

| | | |
|---|---|---|
| 1 | Tags | 4 Historical Data (Includes Data Set Logger) |
| 2 | Alarms and Events | 5 System (Includes Security) |
| 3 | Tag Attributes | 6 DSC Server Development |
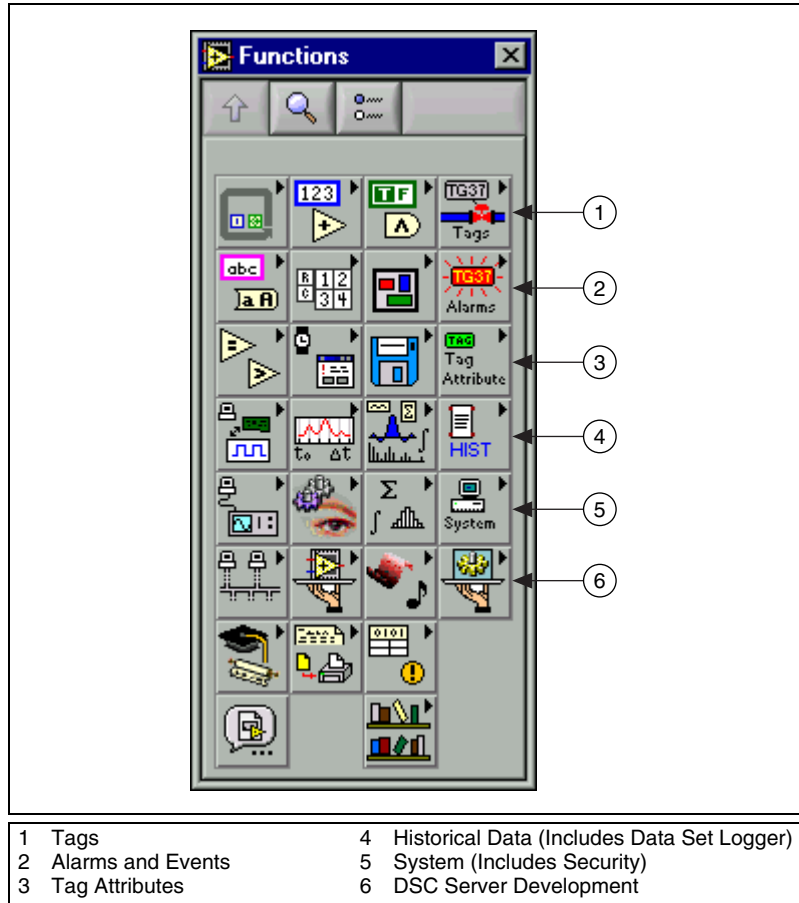
**Figure 1-2.** Functions Palette with the LabVIEW DSC Module

# Utilities

## Tag Utilities Toolbar

Use the **Tag Utilities** floating toolbar to open other LabVIEW DSC module utilities without selecting them through the menus. Select **Tools» Datalogging & Supervisory Control»Show Toolbar** to display the **Tag Utilities** toolbar.

# Tag Configuration Editor

Use the Tag Configuration Editor to create, edit, or delete all of the tags in the LabVIEW DSC module system and to configure Tag Engine parameters. Select **Tools»Datalogging & Supervisory Control»Configure Tags** to open the Tag Configuration Editor.

The Tag Configuration Editor records all tag information and Tag Engine parameters and stores this information in a configuration (.scf) file. The Tag Engine reads this file to determine all of the configuration parameters for execution.

# Tag Monitor

Use the Tag Monitor to monitor the value, timestamp, alarm state, and connection status for selected tags in the system and to write the value to an output or input/output tag. Select **Tools»Datalogging & Supervisory Control»Monitor Tags** to open the Tag Monitor.

# Tag Engine

The Tag Engine runs as a separate application, independent of the HMI application. Both the device servers and the HMI application communicate with the Tag Engine. Select **Tools»Datalogging & Supervisory Control»Launch Engine** to start the Tag Engine.

The Tag Engine performs the following tasks for the LabVIEW DSC module:

• Starts and stops device servers

• Scales and initializes data

• Processes alarms

• Logs alarms and events to the Citadel historical database

• Logs historical data

Servers and the HMI application send data to the Tag Engine. The Tag Engine logs data to the Citadel historical database and maintains the real-time database.

## Citadel Historical Database

Citadel is a National Instruments database used by the LabVIEW DSC module, Lookout, and other National Instruments products. It efficiently stores data acquired and processed by applications.

Refer to Chapter 7, *Historical Data Logging and Extraction*, for more information about the Citadel database.

## Classic Historical Trend Viewer and Historical Data Viewer

Use the classic Historical Trend Viewer (HTV) to view the data stored in any Citadel database. The Historical Data Viewer is another way to view and manage Citadel data. It exists outside of the LabVIEW environment, in the Measurement & Automation Explorer (MAX) environment, and requires no programming. Select **Tools»Datalogging & Supervisory Control»View Historical Data** to open the HTV or Historical Data Viewer.

Refer to the *Extracting and Viewing Historical Data* section of Chapter 7, *Historical Data Logging and Extraction,* for more information about these options.

## Real-Time Database

The real-time database (RTDB) is a snapshot of the current state of all tags defined in the active `.scf` file. The RTDB stores the tag values, status, date, time, and alarm information. When you read and write tags and acknowledge alarms, the Tag Engine updates the RTDB.

## User Account Manager

Use the User Account Manager to set up and edit individual accounts for users and groups of users who use either the LabVIEW DSC module or the applications you create with it. Use the User Account Manager to create an account for a user, assign a password, control how long the password is valid, set the security level for that user, and determine which security group or groups that user belongs to.

Select **Tools»Datalogging & Supervisory Control»Security»Edit User Accounts** to open the User Account Manager.

Refer to the *Creating and Editing User and Group Accounts* section of Chapter 8, *Security*, for more information about the User Account Manager.

## Server Browser

In the LabVIEW DSC module, a *device server* is an application that communicates with and manages I/O devices such as PLCs, remote I/O devices, remote Tag Engines, and data acquisition (DAQ) plug-in devices. These servers read selected input items and write to them on demand. Refer to Chapter 3, *Servers*, for more information about device servers.

Use the Server Browser to see the device servers in a computer and in other computers on the network. You can view server information and display the front panel of VI servers (if the server is running), launch server configuration software for compatible servers, change OPC settings, and unregister a server. Select **Tools»Datalogging & Supervisory Control» Advanced»Server Browser** to open the Server Browser.

## Image Navigator

Use the Image Navigator to browse through a collection of images you can use to create front panels. Select **Tools»Datalogging & Supervisory Control»Image Navigator** to open the Image Navigator.

Refer to the *Adding and Customizing Graphics* section of Chapter 5, *Displaying and Controlling Data*, for more information about the Image Navigator.

# Customizing Your Work Environment

Complete the following steps to customize your work environment and to set startup options. Refer to the LabVIEW documentation for more information about customizing the LabVIEW work environment.

1.  Select **Tools»Datalogging & Supervisory Control»Options** to display the **Options** dialog box.

2.  Select among the options on the **Environment**, **Startup**, and **Advanced** tabs.

    To view descriptions of these options, press <Ctrl-H> and move the cursor over any field.

3.  Click the **OK** button.

# Changing Palette Views

The LabVIEW DSC module also installs a Datalogging & Supervisory Control palette view for the **Controls** and **Functions** palettes. If the Datalogging & Supervisory Control palette view is not active, complete the following steps to change to this palette view.

1. Click the **Options** button in the **Controls** or **Functions** palette toolbar.

2. Select **Datalogging & Supervisory Control** in the **Palette Set** field.

3. Click the **OK** button.

# 2

# Developing an Application

To develop a LabVIEW Datalogging and Supervisory Control (DSC) module application, you must be familiar with LabVIEW. Refer to the LabVIEW documentation for tutorial activities, examples, and general information about LabVIEW.

## Example Applications

Before you develop a LabVIEW DSC module application, you may want to review the Datalogging and Supervisory Control examples. Select **Help»Find Examples**, then browse by category or keyword to find relevant VI examples.

## Parts of a LabVIEW DSC Module Application

A LabVIEW DSC module application contains three parts that work together—the graphical interface or HMI application, the Tag Engine, and various device servers, as shown in the following illustration. The Tag Engine, with any device servers, runs as a separate application independent of the HMI application, as shown in the following illustration.
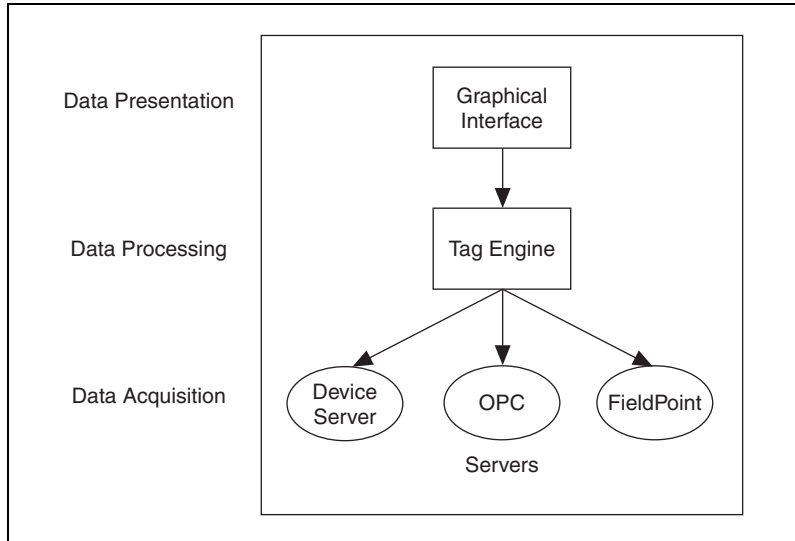
**Figure 2-1.**  Parts of a LabVIEW DSC Module Application

# Basics Steps for Developing an Application

1.  Plan your application requirements, such as channel count, logging, types of channels, hardware, security, and performance. Do not skip this step, because it impacts your decisions at every stage of development including setting up servers, configuring tags, and setting up network clients and servers.

2.  Physically establish a connection to your hardware. Connect, configure, test, and troubleshoot.

3.  Install and configure your device servers to connect to real-world I/O points. This creates the mechanism to handle communication with the hardware.

4.  When possible, before you access servers in LabVIEW, test them according to instructions in the server documentation. Doing so can simplify any necessary troubleshooting.

5.  Use the Tag Configuration Editor to add any additional I/O points and configure logging and alarm settings

6.  Build the HMI application in the same way as a LabVIEW application, taking advantage of the wizard utilities to quickly connect to your I/O and build multi-window front panels.

7.  Use the Tag Monitor to view tag values, timestamps, and alarm states.

8. Use the LabVIEW and LabVIEW Datalogging and Supervisory Control module VIs and functions to add additional functionality to the user interface or HMI application.

9. Deploy, test, and document your application.

# 3

# Servers

In the LabVIEW Datalogging and Supervisory Control (DSC) module, a *device server* is an application that communicates with and manages I/O devices such as PLCs, remote I/O devices, remote Tag Engines, and DAQ plug-in devices. These servers read selected input items and write to them on demand.

The LabVIEW DSC module can connect to any OPC-compliant server and to many third-party device servers. You also can connect to National Instruments servers, including National Instruments DAQ and FieldPoint servers.

A server *item* is a channel, I/O point, or variable in a hardware device. You connect to these items with tags. Device servers monitor the values acquired by the hardware and the Tag Engine updates the tags when the server sends new data to the Tag Engine. Servers also update each output when the HMI application writes that tag value, and they handle and report communications and device errors. A good device server covers all device- and hardware-specific details, thereby establishing a device-independent I/O layer for the LabVIEW DSC module. Many device servers include a configuration utility as well as the run-time application that communicates with the Tag Engine.

When a LabVIEW DSC module application runs, it determines from the configuration (.scf) file which servers are needed and which items are needed from those servers. The LabVIEW DSC module launches each server it needs and monitors the specific items of interest through the Tag Engine.

The LabVIEW DSC module also can function as an OPC server and as a data source for the Logos networking protocol.

A server is not the same as a device driver or an instrument driver. In general, an instrument driver is a software component that is designed to control a programmable instrument such as a multimeter. A device driver is typically a low-level software component that a computer needs to work with a plug-in interface. Such a driver also can function as a server if it meets certain standards, such as the OPC specification.

# Selecting a Server

The LabVIEW DSC module supports several types of servers including the following:

- **OPC servers**—Compliant with version 2.0 of the OPC Data Access specification, as defined by the OPC Foundation.

- **DDE servers**—Any server that supports the Dynamic Data Exchange (DDE) server interface. Refer to the *Using DDE Servers with the LabVIEW DSC Module* section for more information about DDE servers.

- **IA device servers**—A type of server developed by National Instruments. IA device servers have two implementations—VI-based and DLL-based.

    - **VI-based servers**—Use VIs to provide data to the Tag Engine.

    - **DLL-based servers**—Also known as IAK device servers or Industrial Automation Servers (IAS).

You also can use servers provided by a third-party hardware manufacturer. You can build a VI-based server when you have a uniquely configured or custom device that you want to use with the LabVIEW DSC module. Refer to the *Developing Device Servers* section for more information about creating developing VI-based device servers.

## IAK Servers

For those who have used BridgeVIEW and the IAK servers in the past, the LabVIEW DSC module still supports the use of IAK servers. However, National Instruments (NI) recommends upgrading to OPC servers.

To be compatible with future Windows versions, NI recommends that you use OPC servers, specifically the OPC servers for NI-DAQ (including SCXI), National Instruments FieldPoint, and National Instruments FOUNDATION Fieldbus. Refer to the *Notes for BridgeVIEW Users* PDF document installed with the LabVIEW DSC module (`labview6\manuals\bv_note.pdf`), for more information about server compatibility and server strategies for the future.

You can purchase separately the National Instruments Servers CD, which includes the Lookout Protocol Drivers OPC server. You can use the Lookout Protocol Drivers OPC server to connect to a wide variety of devices through the OPC interface protocols. These devices include Allen-Bradley, Siemens, Modbus, and more. The Lookout Protocol Drivers

OPC server also includes a large number of reliable, time-tested, and field-proven Lookout drivers. You can use these Lookout drivers to replace IAK servers, if necessary. The National Instruments Servers CD also includes a patch allowing IAK servers to run in Windows 2000. This patch is provided for backwards compatibility only, without the guarantee that IAK servers can continue to work in future Windows versions.

# Installing and Configuring Servers

After you select the device servers to use with your hardware, install and configure them according to the server documentation.

For many servers, you must use the device server configuration utility to configure how the server monitors items, including how often it polls the devices and other server-specific and device-specific parameters.

## Registering and Unregistering Servers

You might need to register your device servers manually for the LabVIEW DSC module to access them.

### Registering OPC Servers

If a server complies with the OPC specification, it should register itself according to that specification. If an OPC server does not appear in the **Servers** listbox in the Tag Configuration Wizard, refer to the server documentation for information about registering the server.

**Note**    If you change the server registration while the Tag Configuration Editor is open, the change does not appear in the **Servers** listbox. To update the **Servers** listbox while the Tag Configuration Editor is open, select **Servers»Refresh**.

### Registering DDE Servers

You do not need to register DDE servers.

### Registering VI-Based Servers

VI-based servers include a VI you use to register the server. Before you can use a VI-based server, you must run this registration VI.

The LabVIEW DSC module installs VI-based servers, which are the servers used in the LabVIEW DSC module examples. These servers include the tanks server, the SIM server, and the cookie server and are in `labview\examples\lvdsc\servers`.

These servers should already be registered. If they do not appear in the **Servers** listbox in the Tag Configuration Wizard, complete the following steps to manually register the server.

1.  Open the server registration VI in `labview\examples\lvdsc\servers`. For example, open the Register Tanks Server VI to register the tanks server.

2.  Run the VI.

3.  Close the VI.

4.  Repeat steps 1 through 3 for each server you need to register.

If you are writing VI-based device servers, refer to these server registration VIs for examples of registering your servers.

## Unregistering a Device Server

You can unregister an OPC server only by uninstalling the server software.

You can usually unregister VI-based or IAK servers in the Server Browser. Do this only if no tags are configured to use that server and you no longer want to access any items defined by the server. After you unregister a server, you can no longer connect to it from the LabVIEW DSC module and any tag configured to use that server no longer has a valid configuration. After you unregister a device server, you must run the server configuration utility and register it to use it with the LabVIEW DSC module again.

Complete the following steps to unregister a VI-based or IAK device server.

1.  Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2.  Select the server you want to unregister.

3.  Click the **Unregister Server** button.

4.  Click the **Close** button.

Refer to the server documentation for information about unregistering third-party servers.

# Launching Server Configuration Utilities

When you register a VI-based or IAK device server in your system, the LabVIEW DSC module also registers the location of its configuration utility, if possible.

✎ **Note**  On Windows 2000/NT/XP, you might need to log in as an administrator to access server configuration utilities.

Complete the following steps to use the Server Browser to open these same configuration utilities, when available.

1. Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2. Select the server you want to configure in the **Servers** listbox.

3. Click the **Run Server Configuration** button. If no configuration utility is associated with that server, the **Run Server Configuration** button is dimmed.

You also can open registered server configuration utilities from the Tag Configuration Editor by selecting **Servers»***Server Name* **Configuration**, where *Server Name* is the name of the server.

# Viewing Server Information

Use the Server Browser to view information about the device servers in your system and on the network. You also can use the Server Browser to view certain properties of OPC and VI-based servers.

You also can use the Engine Manager to view information about servers in use.

## Viewing Information about All Servers

Complete the following steps to use the Server Browser to view information about all servers.

1. Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2. Select a server in the **Servers** listbox. The symbol to the left of the server name indicates the following information:

   • A black diamond indicates that the server is loaded and running.

   • A white diamond indicates that the server is loaded but not running.

- No symbol indicates that the server is not being used in the current tag configuration.

3. Click the **View Server Information** button. A **Server Information** dialog box appears.

   This dialog box varies based on the type of server you selected and displays general information about the server, devices, and server items. If the Server Browser does not find any devices or items, a checkmark appears in the **No devices found** or **No items found** checkbox, respectively.

   OPC servers have an optional Server Browse Address Space Interface. If a server supports this interface, the LabVIEW DSC module can query it to find which items are available from the server and display them in this dialog box.

4. Select a parameter in the **Sort By** pull-down menu to sort this table by item name, data type, or direction.

## Viewing Information about Running Servers

Complete the following steps to use the Engine Manager to view information about running servers.

1. Open the Engine Manager (with the Tag Engine running, **Tools» Datalogging & Supervisory Control»Launch Engine»Show**).

2. Click the **View Servers in Use** button in the toolbar, shown at left. The **Servers In Use** window appears, listing the servers currently running and supplying data to the Tag Engine.

3. Select a server in the **Server** column. If the server is VI-based, click the **Show** or **Hide** button to show or hide the front panel of the server.

4. Click the **Details** button. A **Server Information** dialog box appears.

   This dialog box varies based on the type of server you selected and displays general information about the server, devices, and server items. If the Server Browser does not find any devices or items, a checkmark appears in the **No devices found** or **No items found** checkbox, respectively.

   OPC servers have an optional Server Browse Address Space Interface. If a server supports this interface, the LabVIEW DSC module can query it to find which items are available from the server and display them in this dialog box.

You can select a parameter in the **Sort By** pull-down menu to sort the information by item name, data type, or direction.

# Connecting Server Items to Tags

After you install, register, and configure a device server, use the Tag Configuration Editor to connect tags to server items. Refer to Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information about creating and configuring tags to read from and write to server items.

# Testing a Server

Complete the following steps to use the Server Browser to make sure your servers are properly installed and configured.

1. Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2. Check the **Servers** listbox to see if your server is listed. If it is not, skip to step 4.

3. Double-click the server to display the **Server Information** dialog box. If the items on that server appear in the dialog box, you successfully installed and configured the server. If the server items do not appear, continue to step 4.

4. Use the configuration utility for that server to check the installation and configuration.

After you configure the tags and save them, complete the following steps to make sure the server is providing data properly.

1. Start the Tag Engine.

2. Open the Tag Monitor.

3. Use the Tag Monitor to connect to tags.

4. Make sure the data values and timestamps change.

5. If you cannot get live data with the Tag Monitor, use the configuration utility for that server to check the installation and configuration.

# Accessing OPC Servers Using the LabVIEW DSC Module as an OPC Client

The LabVIEW DSC module can function as an OPC client and communicate with any server implementing the OPC Foundation OPC server interface, which is a Microsoft COM-based standard. The LabVIEW DSC module finds all OPC servers installed on your computer and reads any available information about the server capabilities and items directly from the server.

## Configuring the LabVIEW DSC Module OPC Client

You might need to configure the LabVIEW DSC module OPC client for the following reasons:

- If the server does not support asynchronous communication, you can force the client to use synchronous communication.

- If the server has many OPC items, you can increase the maximum number of items that display while you browse the server.

- If you perform many writes to the server and receive an output queue overflow, you can increase the length of the OPC write queue, which contains requested but not completed asynchronous writes to the OPC server.

Complete the following steps to configure the LabVIEW DSC module OPC client.

1. Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2. Click the **OPC Client Settings** button.

3. Complete the following steps to use synchronous communications with an OPC server.

   a. Click the **View All** button to display the **View All OPC Servers** dialog box.

   b. Select the server(s).

   c. Click the **Add** button. You also can type the server name in the textbox next to the **Add** button and click the **Add** button.

4. Change the **OPC Items in Browse** value to change the maximum number of OPC items that display while you browse an OPC server.

5. Change the **OPC Write Queue** to change the maximum length of the OPC write queue.

# Accessing Remote OPC Servers through the LabVIEW DSC Module

You can use the LabVIEW DSC module to access OPC servers running on other computers on the network. Use this method to access remote OPC servers so you can access different instances of the same server, such as National Instruments DAQ OPC, running both on your local computer and on the remote computer.

Complete the following steps to use the LabVIEW DSC module to access remote OPC servers.

1. Open the Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**).

2. Click the **Network OPC Servers** button to display the **Browse OPC Servers on Network** dialog box.

   (You could also open the Tag Configuration Editor and select **Servers» Browse Network OPC Servers** to display the **Browse OPC Servers on Network** dialog box.)

3. In the **Network** listbox, navigate to the OPC server you want to access.

4. Select the server and click the **Add Server** button.

   The server appears in the **Registered Remote OPC Servers** listbox. Select a server and click the **Remove Server(s)** button to remove a server from this listbox.

   If you receive an error while connecting to the server, run the `dcomcnfg.exe` Windows utility on the remote computer to configure OPC security options to allow your computer access to launch and connect to the remote computer. For more troubleshooting information, refer to the NI Developer Zone resources at `ni.com/zone`.

# Accessing Remote OPC Servers with dcomcnfg.exe

If necessary, you can use the `dcomcnfg.exe` Windows utility to configure an OPC server to run on a remote computer rather than your local computer. However, if you use `dcomcnfg.exe` to select a remote server, you can run only one version of that server, either locally or on one remote computer. You cannot use the same server on more than one computer. Refer to the *Accessing Remote OPC Servers through the LabVIEW DSC Module* section for more information about accessing different instances of the same server.

Complete the following steps to configure an OPC server to run remotely.

1. Make sure dcomcnfg.exe is installed on your computer.

2. Run dcomcnfg.exe.

3. Click the **Applications** tab.

4. Select the OPC server in the list.

5. Click the **Properties** button to display the **Properties** dialog box.

6. Click the **Location** tab.

7. Remove the checkmark from the **Run application on this machine** checkbox and place a checkmark in the **Run application on the following computer** checkbox.

8. Type the name of the remote computer or click the **Browse** button to navigate to it.

9. Click the **OK** button.

10. Click the **Default Properties** tab and set the following options.

    a. Place a checkmark in the **Enable Distributed COM on this computer** checkbox.

    b. Set the **Default Authentication Level** to **Connect**.

    c. Set the **Default Impersonation Level** to **Identify**.

11. Click the **Default Security** tab.

12. Click the **Edit Default** button. Make sure that the computer on which you want to launch the OPC server is allowed to access your computer. This is necessary for the remote computer to call the LabVIEW DSC module on your computer when supplying OPC values.

# Connecting to Data Published by LabVIEW Real-Time

You can configure a host computer's DataSocket server for use with LabVIEW RT. In this case, LabVIEW RT publishes its real-time data to the DataSocket server on the host machine. You can then use the LabVIEW DSC module to create tags that connect to the data in the DataSocket server running on the host computer.

# Using Other Remote Servers

With the Logos networking protocol, you can access data from any
LabVIEW DSC module application, Lookout application, or FieldPoint
Ethernet module running on computers in your network.

To access LabVIEW and Lookout applications as servers, you must register
the computers on which they are running. Refer to the *Registering
Networked Computers* section of Chapter 9, *Networking and Deploying
Applications*, for more information about registering computers.

# Using DDE Servers with the LabVIEW DSC Module

The LabVIEW DSC module can communicate with any server using DDE
as its interface. A DDE server is a simple server in which you type a device
and item string to select a specific data point to which to connect.

Third-party DDE servers do not register themselves with the LabVIEW
DSC module. Therefore, the LabVIEW DSC module cannot start the DDE
server automatically when it runs the HMI application. To use a DDE
server, start or run the DDE server before you start the Tag Engine. The
LabVIEW DSC module returns system error messages if it cannot connect
to the DDE server when it starts the Tag Engine. Thereafter, it attempts to
reconnect to the DDE server periodically.

# Developing Device Servers

You can develop a VI-based device server to emulate hardware or connect
the LabVIEW DSC module to hardware for which no device server exists.
You can use the same LabVIEW development environment to create the
server as you use to develop your application.

Use the controls located on the **Controls»DSC Server Data Types** palette
and the VIs located on the **Functions»DSC Server Development** palette
to write a VI-based device server.

Refer to the server registration VIs in `labview\examples\lvdsc\`
`servers` for examples of registering your VI-based servers.

The LabVIEW DSC module also installs an Interactive Server Tester utility
you can use to debug VI-based servers. Select **Tools»Datalogging &**

**Supervisory Control»Advanced»Interactive Server Tester** to open the Interactive Server Tester. This tool is useful only with VI-based servers.

Refer to `viserver.pdf` in `labview\manuals` for information about building a VI-based device server and using the Interactive Server Tester.

# 4

# Using Tags to Manage I/O in LabVIEW

In the LabVIEW Datalogging and Supervisory Control (DSC) module, you use a *tag* to create and maintain a connection to a real-world I/O point. You also can use a *memory tag* for data held by your application that you need to use or track. A *network tag* is a tag remotely connected to any type of tag on another Tag Engine.

The tasks you perform through tags depend on how you configure the tag attributes. Tag attributes include how the data a tag reports are scaled; whether, where, and how a tag is logged to a historical database; and alarm levels and priorities for the tag data.

In addition, you can organize tags into logical groups for convenience and efficiency, configure the tag data type, set initialization values, set separate deadbands for logging or updating data, attach units of measurement to data, attach an alarm message to a tag whose values enter the alarm ranges you set, and set alarm deadbands separate from the logging and update deadbands.

You perform tag management in the Tag Configuration Editor, which you access by selecting **Tools»Datalogging & Supervisory Control» Configure Tags**. Before you create or configure tags, you must install and configure your servers. Refer to Chapter 3, *Servers*, for more information about installing and configuring servers.

## Configuration Files

After you create tags and configure their attributes, you save that information in a configuration (.scf) file. Any LabVIEW DSC module utility that needs tag information uses the .scf file. These utilities include the Tag Engine, Tag Monitor, and HMI Wizard, which generally access the .scf file to find a list of active tags and other configuration information.

The `.scf` file does not contain any information about the VIs in the HMI. In fact, it does not need to be specific to any single application. Multiple applications can run concurrently if they use the same `.scf` file.

# Changing the Active SCF File

With the exception of the first time you run the LabVIEW DSC module, the active (default) `.scf` file is the last `.scf` file you saved with the Tag Configuration Editor. The Tag Configuration Editor opens the active `.scf` file by default and the Tag Engine accesses the active `.scf` file by default. You can change the active `.scf` file manually or programmatically.

## Changing the Active SCF File Manually

Complete the following steps to change the active `.scf` file manually.

1.  Select **Tools»Datalogging & Supervisory Control»Options**.

2.  Click the **Environment** tab.

3.  Change the **Default SCF** value.

    This change could take effect immediately, depending on the options you select in the **Options** dialog box.

## Changing the Active SCF File Programmatically

Complete the following steps to change the active `.scf` file programmatically in a VI.

1.  Use the Load SCF VI or Engine Launch VI located on the **Functions»System** palette.

2.  Connect the `.scf` filename you want to the **Configuration File to Use** input terminal.

# Creating Tags

You can create tags in several ways:

•   Generate tags automatically in the Tag Configuration Wizard.

•   Create tags manually in the Tag Configuration Editor.

•   Create tags in the HMI Wizard before you connect front panel objects to a tag value.

**Note**   You must create DDE server connections manually in the Tag Configuration Editor, instead of in the Tag Configuration Wizard.

# Generating Tags Automatically

Use the Tag Configuration Wizard to generate tags from the server information if you want the Tag Engine to monitor a large number of I/O points in your system. When you run the server configuration utilities for the servers on your system, you can define devices and items for the I/O points that the servers monitor and control. You can then generate tags from these server items in the Tag Configuration Wizard.

The wizard uses the tag name, data type, I/O group, I/O connection, and scaling attributes for each server item to create the tags. For IAK and VI-based servers, the wizard reads server information from the Common Configuration Database (CCDB). For OPC servers that support the Server Browse Address Space Interface, the wizard reads server information by browsing the server address space. The wizard uses the default tag attributes to configure the remaining attributes. You can change the default tag attributes in the Tag Configuration Wizard by clicking the **Set Tag Defaults** button. Refer to the *Defining Default Values for Tag Configuration Fields* section for more information.

**Note**   You must create DDE server connections manually in the Tag Configuration Editor, instead of in the Tag Configuration Wizard. Refer to the *Creating Tags Manually* section for more information about creating tags in the Tag Configuration Editor.

Complete the following steps to use the Tag Configuration Wizard to generate tags.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Click the **Configuration Wizard** button in the toolbar, shown at left.

3. Expand each server branch in the **Servers** tree view to display the devices and items for one or more servers. If the **Servers** tree view lists item ranges instead of individual item names, skip to step 7.

4. Select the items for which you want to create tags.

   Select a branch in the **Servers** tree view to generate tags for all the items in that branch. Select a server branch to generate tags for all items on that server.

5. (Optional) You can create a tag for a DataSocket item:

   a. Click the **DataSocket** button.

   b. Double-click on **DataSocket Server** to see the items on your local machine, or browse to a networked machine under **Network Neighborhood** and view its DataSocket items.

c.  Select the item for which you want to create a tag. Currently, you can create only one tag at a time. The tag created from this item will use the entire URL as its tag name (the "/" will be replaced with "_"), with the data type and access rights shown.

d.  Click **OK**.

6.  Click the **Add Item(s)** button. The Tag Configuration Wizard moves the selected items to the **Selected Items** listbox.

7.  (Optional) Some OPC servers do not list individual item names in their hierarchical tree, but instead provide ranges for item names. This is common when the server contains a large set of items. These item ranges help you create specific item names. The format for the item ranges depends on the OPC server. If your server uses item ranges, complete the following steps to generate tags from an item range.

a.  Select an item range.

b.  Click the **Add as Range** button to display the **Add Items in a Range** dialog box.

c.  Type the starting item name and set the number of items that you want to create.

The Tag Configuration Wizard creates the item names, incrementing the trailing numbers in the starting item name. If you did not add a trailing number to the starting item name, the Tag Configuration Wizard appends a zero to the first name and increments trailing numbers in each subsequent name.

8.  (Optional) The Tag Configuration Wizard uses the tag configuration defaults to set most of the tag parameter values. To change these defaults, click the **Set Tag Defaults** button.

9.  (Optional) The Tag Configuration Wizard automatically creates I/O groups for each server and uses the tag configuration defaults to set the I/O group rate and deadband settings. The Tag Configuration Wizard also sets the I/O group name to the server name. Complete the following steps to change the I/O group settings for each server.

a.  Select a server.

b.  Click the **Properties** button to display the **Properties of Tags Generated for Device/Server** dialog box.

c.  Click the **I/O Group** tab.

d.  Select among the I/O group settings.

10. (Optional) The Tag Configuration Wizard sets the tag name to the item name for each tag created. For non-OPC servers that have devices, the tag name contains both the device and item name if the server has more than one device. Follow these steps if you want to change the tag name format for a server.

    a.  Select a server.

    b.  Click the **Properties** button to display the **Properties of Tags Generated for Device/Server** dialog box.

    c.  Click the **Tag Names** tab.

    d.  Set the tag name format.

11. To remove individual items from the **Selected Items** listbox, select the items and click the **Remove Item(s)** button. To remove all of the items, click the **Remove All** button.

12. When all the items for which you want to create tags are in the **Selected Items** listbox, click the **OK** button.

    The Tag Configuration Editor creates tags for each item and appends the tags to the current tag configuration (.scf) file.

13. (Optional) If you want the changes to be a separate .scf file, select **File»Save As** and save the file with a different name.

## Creating Tags Manually

When you generate tags, you can either add them to an existing configuration, or you can create a new configuration file (.scf). You can later manually change the configuration of any tag.

1.  If you have not already done so, install and configure your server(s) as described in Chapter 3, *Servers*.

2.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

3.  Select **Edit»Create** and select the type of tag you want to create. A **Tag Configuration** dialog box appears.

4.  Select among the tag configuration options. The tag name must be unique within a given .scf file. Refer to the *Configuring Tag Attributes* section for more information about the tabs and fields in this dialog box.

5.  Click the **OK** button to create the new tag or click the **Create Next Tag** button to create the new tag and create another tag of the same type.

6.  Select **File»Save** to save the changes.

## Importing Network Tags

You can import tags into a local `.scf` file from a `.scf` file located on another computer.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Select **File»Import Network Tags** to display the **Select Tags for Network Import** dialog box.
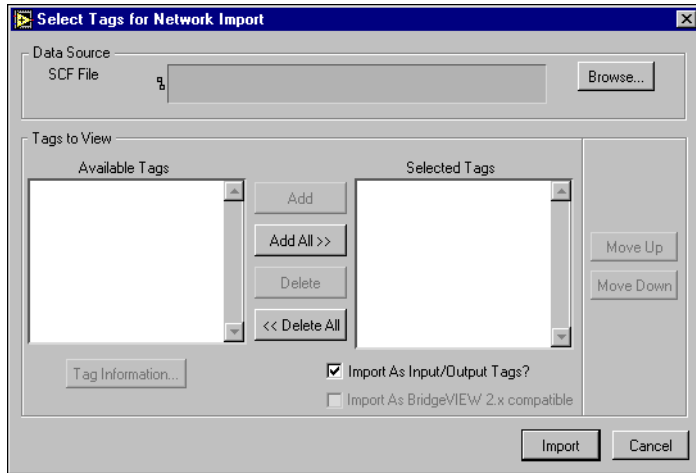


3.  Click the **Browse** button next to the **SCF File** field and navigate to a `.scf` file on any computer on the network.

4.  Click the **Add** buttons to add the tags you want to import to the **Selected Tags** listbox.

5.  (Optional) To import tags from a remote computer using BridgeVIEW 2.*x* networking, place a checkmark in the **Import As BridgeVIEW 2.x compatible** checkbox. To convert to the Logos networking protocol, do not place a checkmark in this checkbox. National Instruments recommends converting to the Logos networking protocol when possible, but the BridgeVIEW 2.*x* networking option is provided in case you need backward compatibility. Refer to the NI Developer Zone resources at `ni.com/zone` for more information about networking options.

6.  Click the **Convert** button to import tags from that file into the local `.scf` file.

## Importing Virtual DAQ Channels as Tags

You can create memory tags that use DAQ virtual channel names to incorporate LabVIEW DSC module tags into your existing VI-based DAQ application. Using this method, you can take advantage of the LabVIEW DSC module features such as alarming, logging, and security, without reconfiguring your virtual channels. Follow these steps to import DAQ virtual channels as tags.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select **File»Import DAQ Memory Tags**.

3. Click **Add All** or select the DAQ channels you want to import and click **Add**.

4. Click **OK**. You should see memory tags with the same names as your DAQ channels.

## Creating Tags from the HMI Wizard

You can create a tag when you right-click a front panel control or indicator and run the HMI Wizard. Refer to the *Connecting Controls and Indicators to Tags* section of Chapter 5, *Displaying and Controlling Data*, for more information.

# Editing Tags

You can edit tags manually, programmatically, or in a spreadsheet.

## Editing Tag Configuration Manually

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click the tag to display the **Tag Configuration** dialog box.

3. Select among the tag configuration options. Refer to the *Configuring Tag Attributes* section for more information about the tabs and fields in this dialog box.

4. Click the **OK** button. A diamond appears next to the tag to indicate it has changed.

5. Select **File»Save** to save the changes.

If you want to update the Tag Engine and any static attributes have been changed, the Tag Engine shuts down and restarts. If you have changed only dynamic attributes in the .scf file, the Tag Engine is updated without restarting.

⚠  **Caution**    Communication between the Tag Engine and any device server is stopped temporarily when the Tag Engine shuts down and restarts.

## Editing Tag Configuration Programmatically

You can edit tag configuration information programmatically, which is helpful when displaying engineering units, scales, and other information about the tag or changing a tag from Offscan to Onscan. Use the Tag Attributes VIs to edit tag configuration information programmatically. Refer to the *Changing Tag Attributes Programatically* section of Chapter 5, *Displaying and Controlling Data*, for more information about the Tag Attributes VIs.

## Editing Tag Configuration in a Spreadsheet

With the Tag Configuration Editor, you can export tag configuration information to spreadsheet files and import tag configuration information from spreadsheet files. The files are tab-delimited text (.txt) files.

If you use spreadsheet files with the Tag Configuration Editor, it is important that you consider the following:

- If you do not select *all* of the fields when exporting data, you lose configuration information when you import it back to the Tag Configuration Editor.

- You can export a subset of information, and then rely on tag default parameters when you import the data back into the Tag Configuration Editor. However, each row in the spreadsheet file must contain the tag name and data type fields, or the import mechanism cannot read it.

- Some configuration parameters, such as those in the **Historical Logging Configuration** and **Event Configuration** dialog boxes, are inherited from the currently open .scf file when you import spreadsheet data.

- When importing, you can append the imported tags to the current .scf file.

- If you create a spreadsheet file to import as a tag configuration, use the same format as a file created by exporting an existing tag configuration.

## Exporting Tag Configuration to a Spreadsheet

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select **File»Export** to save the file as a tab-delimited `.txt` file.

3. A dialog box prompts you to select and order the fields you want in the spreadsheet file. If you want to edit the spreadsheet and import the edited data back into the Tag Configuration Editor, click the **All** button to select all available fields. Click the **Default Order** button to restore the order of the fields to the default order.

## Importing Tag Configuration from a Spreadsheet

1. Save the spreadsheet as a `.txt` file.

2. In LabVIEW, open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

3. (Optional) Select **File»New** to open a new `.scf` file for the imported tag configuration.

4. Select **File»Import** and select the `.txt` file to import the data from the spreadsheet.

5. Select **File»Save** to save the changes.

# Defining Default Values for Tag Configuration Fields

You can simplify the tag configuration process by defining default values for several fields. These default values are then used when you create tags automatically, such as with the Tag Configuration Wizard or by importing. For example, you might want to set the default to **Log Data** or **Log Events**, or set the log deadband to a particular value by default.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select **Configure»Default Parameters** to display the **Set Default Parameters** dialog box. You also can click the **Set Tag Defaults** button in the Tag Configuration Wizard to display this dialog box.

3. Set the default values for the parameters listed. Refer to the *Configuring Tag Attributes* section for more information about these parameters.

4. Click the **OK** button.

The default values apply when creating a new tag, importing a tag from a server, or importing a tag from a spreadsheet. In the case of a spreadsheet, a value in the spreadsheet overrides the default value for the field.

# Setting Tag Deadbands

A deadband is a filter that eliminates noise from data. Any changes in value from a data point are compared to the previous value, and only if the difference between the new value and the previous value exceeds the deadband does the new value replace the old. Deadbands in LabVIEW DSC module tags are set as a percentage of the value range of a data point.

The Tag Engine uses update deadband and log deadband values to eliminate unnecessary processing on insignificant data value changes. Deadband allows you to define what constitutes a significant change. The Tag Engine ignores an operation if the change in data is not considered significant. You can increase engine throughput by increasing deadband size (though this might compromise data resolution). If you set the update deadband too high, the RTDB might not be updated. This might result in inadequate historical logging or alarm management. In addition, you can configure a server to apply a deadband to items associated with an I/O group.

## Deadbanding Interaction

Three deadband settings are directly under your control when you configure individual tags. The fourth takes advantage of an OPC specification and is consequently subject to how that specification was implemented in a given OPC server.

The three deadband settings that are directly under your control are update deadband, log deadband, and alarm deadband.

The update deadband affects how the Tag Engine updates values in the RTDB. The log and the alarm deadbands both operate on the values contained in the RTDB, that is, values that have passed through the update deadband. If the update deadband is set too wide, under some circumstances, it can interfere with your intended alarm and log settings. A careful analysis of the interaction of your deadbands might be indicated, depending on how critical your requirements are.

The deadband setting that takes advantage of an OPC specification is the OPC server I/O group deadband. This deadband is implemented in the server. This deadband affects values coming from the server before the Tag Engine gets the value, so the effects of a deadband setting here can ripple through the update deadband and the log and alarm deadbands. Also, because items in an I/O group can have different ranges, the percentage you select as a deadband might have different numeric results with different items. Refer to your server documentation before you change OPC server I/O group deadband settings.

# Setting Update Deadbands

When you set an update deadband, any new value acquired by the Tag Engine is compared to the existing value held in the RTDB. The new value replaces the existing value in the RTDB only when the difference between the new value and the existing value exceeds the update deadband. You set the update deadband in the tag configuration **Operations** tab.

## Example

For a data point with a range of values of 0 to 100, set the update deadband to 1%. The existing value in the RTDB is 12.3. If the Tag Engine reports a new value of 13, the RTDB does not update because the change in value did not exceed the deadband. If the Tag Engine reports a new value of 11, the RTDB updates because the difference is greater than the deadband.

# Setting Log Deadbands

When you set a log deadband, the new value in the RTDB is compared to the old value in the RTDB. The new value is logged if it exceeds the log deadband. You set the log deadband in the tag configuration **Operations** tab.

The default setting for the log deadband is 1%.

## Example

For a data point with a range of values of 0 to 100, set the logging deadband to 2%. The last value logged was 12.3. When the RTDB updates to 11, the updated value is not logged because it is smaller than the deadband. The value in the RTDB must be greater than 14.3 or less than 10.3 for the data point to be logged.

## Setting Alarm Deadbands

When you set an alarm deadband, the new value in the RTDB is compared to the old value in the RTDB. The alarm is triggered when the value falls outside the range of the deadband and is cleared when it reaches the inside range of the deadband. You set the alarm deadband in the tag configuration **Alarm** tab.

### Example

For a data point with a range of values of 0 to 100, set a LO condition alarm at a value of 12 with a deadband of 1.5%. Your alarm condition is not triggered to active until the RTDB value drops to 12 or below. The alarm stays active until the RTDB value rises to 13.5 or greater.

## Setting I/O Group Deadbands with OPC Servers

When you set a deadband for I/O groups in OPC servers, the OPC server gets the tag value, then filters the tags in the I/O group by deadband before sending the data to the Tag Engine. The Tag Engine can filter data by deadband again before sending the values to the RTDB.

Complete the following steps to set I/O group deadbands.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Double-click a tag in the I/O group for which you want to set deadbands. The **Tag Configuration** dialog box appears.

3.  Click the **Connection** tab.

4.  Click the **Edit** button located under the **I/O Group** pull-down menu to display the **IO Group Configuration** dialog box.

5.  Change the **Update Rate** and **Deadband** values.

6.  Click the **OK** button twice.

The percentage you set applies to the range of each individual OPC item, so the actual raw value of the deadband might change from item to item. This I/O group deadband takes place in the OPC server. Settings made in the OPC server might impact the effect of your deadband setting. Refer to your OPC server documentation for more information about that server.

# Deleting Tags

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select the tag(s) you want to delete.

3. Click the **Delete Tag** button on the toolbar. A trash can icon appears next to the tags.

4. Select **File»Save** to delete the marked tags. The Tag Configuration Editor removes the tag and its configuration information from the .scf file. You still can retrieve historical and event information about the tag, but the Tag Configuration Editor removes information such as the tag description, units, range, and alarm settings.

You can use the **Delete Tag** button in the button bar of the Tag Configuration Editor to undelete tags if all selected tags have a trash can symbol.

# Configuring Tag Attributes

Tag attributes set how the Tag Engine handles a tag. There are five categories of tag attributes: General, Connection, Operations, Scaling, and Alarms.

When you create a tag using the Tag Wizard, the Tag Wizard assigns the default values for each tag attribute. Refer to the *Defining Default Values for Tag Configuration Fields* section for more information about setting tag default values.

When you create a tag manually by selecting **Edit»Create** in the Tag Configuration Editor, you can set each attribute in the **Tag Configuration** dialog box that appears.

Complete the following steps to edit the attributes of an existing tag.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click the tag to display the **Tag Configuration** dialog box.

3. Click the following tabs and select among the options.

   • **General**—Attributes such as tag name, group, and description.

   • **Connection**—Attributes that describe where the Tag Engine sends or receives values for the tag and how to access that data.

These tags have access rights of input, output, or input/output.
Memory tags are not connected to a real-world I/O point; set their
**Tag Access** to **Memory**. Refer to the *Determining When to Use
Memory Tags* section for more information about memory tags.

- **Operations**—Attributes that describe additional functionality
  that the Tag Engine performs on a tag or its values.

- **Scaling**—Attributes that describe which scaling function is
  applied to a tag value.

- **Alarms**—Attributes that describe abnormal process conditions
  for a given tag.

4. Click the **OK** button.

5. Select **File»Save** to save the changes.

6. Use the Tag Monitor to test the tag configuration and make sure you
   are reading and writing data properly with your servers. Refer to the
   *Monitoring and Writing Tag Values* section for more information
   about using the Tag Monitor.

# Tag Data Type

How you configure a tag varies slightly depending on the data type. Tag
data types include analog, discrete, bit array, and string tags.

## Analog Tags

An *analog tag* is a continuous value representation of a connection to
a real-world I/O point or memory variable. This type of tag can vary
continuously over a range of values within a signal range.

Use an analog tag when you want to express a continuous value
(for example, 0 to 100).

## Discrete Tags

A *discrete tag* (Boolean control or indicator in LabVIEW) is a two-state
(ON/OFF) value representation of a connection to a real-world I/O point
or memory variable. This type of tag can be either a 1 (TRUE) or a
0 (FALSE).

Use a discrete tag when you want to express a two-state (ON/OFF) value.

## Bit Array Tags

A *bit array tag* is a multi-bit value representation of a connection to a real-world I/O point or memory variable. This type of tag can be comprised of up to 32 discrete values.

Use a bit array tag when you have a multi-bit value in which each of the bits represents a flag or single value that is turned on or off. The maximum length of a bit array tag is 32.

LabVIEW stores a bit array as a number (which is what displays in the Tag Monitor), but it is an array of bit values.

## String Tags

A *string tag* is an ASCII or binary character representation of a connection to a real-world I/O point or memory variable.

Use a string tag when you have binary information or an ASCII value. When you configure a string tag, you must select whether to treat the data in the tag as text or binary information. You might use a string tag to obtain values from a bar code reader, or if you have data that does not fit into any other data type. You also can use a string tag for PLC control strings and PLC reporting strings.

# Static and Dynamic Attributes

Tag attributes are classified as either static or dynamic attributes. *Static attributes* require you to restart the Tag Engine when you change them in the Tag Configuration Editor. A static attribute change is marked with a solid diamond in the Tag Configuration Editor. Examples of static attributes include general attributes and I/O connection attributes, such as server, device, or item.

*Dynamic attributes* do not require the Tag Engine to restart. The Tag Configuration Editor can change a dynamic tag attribute in a running Tag Engine. A dynamic attribute change is marked with a hollow diamond in the Tag Configuration Editor. Examples of dynamic attributes include enabling logging operations, alarm attributes, and some scaling attributes.

## Defining a Tag Group

Use tag groups to define a subset of tags in the system. You also can use tag groups to examine the alarm states for a subset of tags in the system. Refer to Chapter 6, *Alarms and Events*, for more information about alarm groups.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **General** tab.

4. Select an existing tag group or define a new tag group by selecting **Enter New**.

5. (Optional) To view the tag groups, select **Configure»Tag Groups**. The **Tag Group Display** dialog box appears. Click the **Remove Tag Group** button to delete the tag group.

6. Click the **OK** button.

7. Select **File»Save** to save the changes.

## Configuring I/O Groups

I/O groups are used to configure item rate and deadband for items of a server and to select a specific device, if the server uses devices. For servers that support resource configuration, you also can use I/O groups to configure devices and communication resources. For OPC servers, an I/O group conforms to the concept of an OPC group, which is user-defined and controls timing. Each I/O group you create maps to an OPC group in the OPC server with the same attributes. An I/O group is associated with only one server and, if that server uses devices, with only one device. A server can have multiple I/O groups associated with it.

Any tag other than a memory tag must be part of an I/O group. If you are editing a tag an I/O group probably already exists. If you want the tag to go into a new I/O group or your are creating a tag, you will have to create an I/O group before connecting your tag to a server item.

Complete the following steps to edit the I/O group configuration.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click the tag to display the **Tag Configuration** dialog box.

3. Click the **Connection** tab.

4.  Click the following buttons, which are located under the **I/O Group** pull-down menu. If the **I/O Group** pull-down menu is dimmed, set **Tag Access** to something other than **Memory**.

    •  To create an I/O group, click the **Create** button. Select a device from the list of available devices that appears.

       A list of items connected to that device appears in the **Tag Configuration** dialog box. For a selected device and item, the Tag Configuration Editor imports any available item engineering range and unit information and also makes sure the directions or access rights for an item are compatible with the access rights you have selected for the tag.

       If a device server does not appear in the server name list, you must run the configuration or registration utility for your server before the LabVIEW DSC module can access the server. Refer to the *Installing and Configuring Servers* section of Chapter 3, *Servers*, for more information.

    •  To edit an I/O group, select a group in the **I/O Group** pull-down menu and click the **Edit** button.

    •  To delete an I/O group, select a group in the **I/O Group** pull-down menu and click the **Delete** button. The I/O group is deleted from the server configuration. Deleting an I/O group does not delete the device and communication resource from the server configuration.

5.  Click the **OK** button.

6.  Select **File»Save** to save the changes.

## Configuring DDE Devices and Items

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Double-click the tag to display the **Tag Configuration** dialog box.

3.  Click the **Connection** tab.

4.  Click the **Create** or **Edit** button under the **I/O Group** pull-down menu. The **IO Group Configuration** dialog box appears.

5.  Click the **Add** button under the **Device** pull-down menu. If no **Add** button is available, you are not configuring a DDE device or item.

6. Type the DDE application name and topic in the form
   *application*|*topic* in the **Device** textbox. For example, type
   excel|*worksheetname* to connect to a cell in Microsoft Excel.

   If you are using network DDE to use a DDE server running on another
   computer, use the network DDE name for the *application* part of
   the device name. Refer to your DDE server documentation for more
   information about application and topic names.

7. Click the **OK** button twice.

8. Click the **Add** button under the **Access Path** pull-down menu.

9. Type the name of the item you want to connect to in the **Item** textbox.
   For example, type r2c2 to connect to cell B2 in Excel. You cannot
   browse a DDE server for available items. Refer to your DDE server
   documentation for more information about item names.

10. Click the **OK** button.

11. Select **File»Save** to save the changes.

## Configuring Communication Resources

You can configure communication resources only for IAK servers.
Complete the following steps to configure the communication resource.

1. Open the Tag Configuration Editor (**Tools»Datalogging &
   Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **Connection** tab.

4. Click the **Create** or **Edit** button under the **I/O Group** pull-down menu.
   The **IO Group Configuration** dialog box appears.

5. Click the following buttons, which are located under the
   **Communication Resource** pull-down menu when configuring an
   IAK server.

   • To create a new communication resource, click the **Create** button.

   • To edit a communication resource, select a resource in the
     **Communication Resource** pull-down menu and click the **Edit**
     button.

   • To delete a communication resource, select a resource in the
     **Communication Resource** pull-down menu and click the **Delete**
     button.

6. Click the **OK** button twice.

7. Select **File»Save** to save the changes.

# Configuring Device Names

You can configure device names only for servers that allow users to
configure device names, such as DDE servers. DDE servers use the
device name to specify the DDE application and topic. Complete the
following steps to configure the device name.

1. Open the Tag Configuration Editor (**Tools»Datalogging &
   Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **Connection** tab.

4. Click the **Create** or **Edit** button under the **I/O Group** pull-down menu.
   The **IO Group Configuration** dialog box appears.

5. Click the following buttons, which are located under the **Device**
   pull-down menu for servers that allow you to configure device names.

   • To add a device name, click the **Add** button. In the **Add Device
     Name** dialog box, enter a new device name for a server and click
     the **OK** button.

   • To edit a device name, select a device in the **Device** pull-down
     menu and click the **Edit** button. In the **Edit Device Name** dialog
     box, edit the existing device name for a server and click the
     **OK** button.

   • To delete a device name, select a device in the **Device** pull-down
     menu and click the **Delete** button. The selected device name is
     removed from the device list.

6. Click the **OK** button.

7. Select **File»Save** to save the changes.

# Configuring Device Resources

You can configure device resources only for servers that allow users to
configure device resources. Complete the following steps to configure the
device resources.

1. Open the Tag Configuration Editor (**Tools»Datalogging &
   Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **Connection** tab.

4. Click the **Create** or **Edit** button under the **I/O Group** pull-down menu.
   The **IO Group Configuration** dialog box appears.

5.  Click the following buttons, which are located under the **Device** pull-down menu for servers that allow you to configure device resources. The options in the **Device Configuration** dialog box vary depending on the type of server.

    •   To create a new device configuration, click the **Create** button. In the **Device Configuration** dialog box, configure the device and click the **OK** button.

    •   To edit a device configuration, select a device in the **Device** pull-down menu and click the **Edit** button. In the **Device Configuration** dialog box, edit the existing device configuration and click the **OK** button.

    •   To delete a device configuration, select a device in the **Device** pull-down menu and click the **Delete** button. The selected device name is removed from the server configuration.

6.  Click the **OK** button.

7.  Select **File»Save** to save the changes.

## Configuring Item Names

You can configure item names only for servers that allow users to configure item names, such as DDE servers. Complete the following steps to configure item names.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Double-click a tag to display the **Tag Configuration** dialog box.

3.  Click the **Connection** tab.

4.  Click the following buttons, which are located under the **Access Path** pull-down menu. If the server does not support item names, these buttons are disabled.

    •   To add an item name, click the **Add** button. In the **Add Item Name** dialog box, enter a new item name for a selected server and click the **OK** button.

    •   To edit an item name, select an item in the **Item** pull-down menu and click the **Edit** button. In the **Edit Item Name** dialog box, edit the existing item name for a selected server and click the **OK** button. If the server has access paths, you also can edit an access path.

        For OPC servers, you also can click the **Browse** button to view the hierarchical organization of the server items, navigate to an item, select it, click the **OK** button, and click the **Edit** button.

- To delete an item name, select an item in the **Item** pull-down menu and click the **Delete** button. The selected item name is removed from the item list. If the server has access paths, the selected access path is removed from the access path list.

5. Click the **OK** button.

6. Select **File»Save** to save the changes.

## Configuring Item Resources

You can configure item resources only for servers that allow users to configure item resources, such as many OPC servers. Complete the following steps to configure item resources.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **Connection** tab.

4. Click the following buttons, which are located under the **Access Path** pull-down menu. If the server does not support item configuration, or if the selected item is not valid, these buttons are disabled.

   - To create an item resource, click the **Add** button. In the configuration dialog box, configure a new item for a selected server and click the **OK** button.

   - To edit an item resource, select an item in the **Item** pull-down menu and click the **Edit** button. In the server-dependent dialog box, edit the configuration of the selected item and click the **OK** button

   - To delete an item resource, select an item in the **Item** pull-down menu and click the **Delete** button. The selected item is removed from the server configuration.

   - To browse available items from OPC servers that support browsing, click the **Browse** button. In the **Browse OPC Server** dialog box, browse the list of available items, select an item and associated access path, and click the **OK** button

   - To use the item name as the tag name, click the **Paste Item Name to Tag Name** button. Clicking this button replaces any name in the **Tag Name** field on the **General** tab.

5. Click the **OK** button.

6. Select **File»Save** to save the changes.

## Configuring a Tag to Log Data or Events

By default, when a tag is first created, the LabVIEW DSC module enables logging, so when you start logging, you log all tags except those you have configured not to be logged. To start logging, you either manually activate it in the Engine Manager or you set logging to begin automatically when the Tag Engine starts.

To enable logging programmatically, use the System VIs that enable event or historical datalogging.

Complete the following steps to configure logging manually.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click the tag to display the **Tag Configuration** dialog box.

3. Click the **Operations** tab.

4. Check or uncheck the **Log/Print Events** and **Log Data** checkboxes. Events, in this case, include enabled alarms for the tag. Set the logging deadband and the data resolution, if applicable.

5. Click the **OK** button.

6. Select **File»Save** to save the changes.

7. Make sure the Tag Engine is set to log historical data or events. Refer to *Logging Historical Data* in Chapter 7, *Historical Data Logging and Extraction*.

The LabVIEW DSC module logs data from all tags that have been configured for logging.

## Setting Startup Tag Values

Set startup values to initialize a tag to a known value when the Tag Engine starts.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click a tag to display the **Tag Configuration** dialog box.

3. Click the **Operations** tab.

4. Check the **Set Initial Value** checkbox.

5. Type the initial value in the adjacent textbox.

6. Click the **OK** button.

7. Select **File»Save** to save the changes.

# Scaling Tags

Scaling is useful for converting the range of values from measured units into a calculated range. Only analog (numeric), discrete, and bit array tags have scaling attributes. There is no scaling for string tags or memory tags.

Often an application needs the LabVIEW DSC module to manipulate the raw data used in the device server to put it in a form, called engineering units, suitable for the operators.

## Scaling Analog Tags

You can define the raw range and engineering range for a tag to perform simple conversions between the two ranges. The raw range, defined by Raw Full Scale and Raw Zero Scale, refers to the values used by the device server. Engineering range, defined by Engineering Full Scale and Engineering Zero Scale, refers to the values used by the Tag Engine and HMI.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Double-click an analog tag to display the **Tag Configuration** dialog box.

3.  Click the **Scaling** tab.

4.  Select **Linear** in the **Scale Type** pull-down menu to enable a linear conversion between raw and engineering ranges, *(mx + b)*. Select **Square Root** to enable a square root conversion between the raw and engineering ranges, which is *b + m \* sqrt(raw − o)* where *b* = EngMin, *m* = (EngMax − EngMin)/sqrt(RawMax-RawMin), and *o* = RawMin.

5.  Click the **OK** button.

6.  Select **File»Save** to save the changes.

### Square Root and Linear Scaling

Linear scaling is a straight proportional scale of raw values to engineering unit values.

Square root scaling is a proportional way of scaling raw values to engineering units. It is generally used when scaling down by scaling to the square root of the of the raw unit (after compensating for any offsets involved).

Suppose you want to scale a raw value that ranges from 0 to 100 to engineering units ranging from 0 to 10. The tag returns values, as shown in the Table 4-1.

**Table 4-1.** Square Root and Linear Scaling Example Values

| Raw Units | Linear Scale | Square Root Scale |
|:---:|:---:|:---:|
| 0 | 0 | 0 |
| 4 | .4 | 2 |
| 9 | .9 | 3 |
| 16 | 1.6 | 4 |
| 25 | 2.5 | 5 |
| 36 | 3.6 | 6 |
| 49 | 4.9 | 7 |
| 81 | 8.1 | 9 |
| 64 | 6.4 | 8 |
| 10 | 1 | 3.16 |
| 20 | 2 | 4.47 |
| 30 | 3 | 5.48 |
| 40 | 4 | 6.32 |
| 50 | 5 | 7.07 |
| 60 | 6 | 7.75 |
| 70 | 7 | 8.37 |
| 80 | 8 | 8.94 |
| 90 | 9 | 9.49 |
| 100 | 10 | 10 |

Table 4-2 shows a raw value that ranges from 0 to 100 scaled to engineering units ranging from 15 to 30. Offsets deliver somewhat more complicated results.

**Table 4-2.** Scaling with Offset Example Values

| Raw Units | Linear Scale | Square Root Scale |
|:---------:|:------------:|:-----------------:|
| 0 | 15 | 15 |
| 4 | 15.60 | 18 |
| 16 | 17.40 | 21 |
| 36 | 20.40 | 24 |
| 64 | 24.60 | 27 |
| 10 | 16.50 | 19.74 |
| 20 | 18 | 21.71 |
| 30 | 19.50 | 23.22 |
| 40 | 21 | 24.49 |
| 50 | 22.50 | 25.61 |
| 60 | 24 | 26.62 |
| 70 | 25.50 | 27.55 |
| 80 | 27 | 28.42 |
| 90 | 28.5 | 29.23 |
| 100 | 30 | 30 |

### Example—Linear Scaling

A device server returns a simple voltage from 0 to 5 V. The voltage is related to a position sensor, and the real-world position is measured in centimeters, with 0 volts mapped to 50 cm and 5 V mapped to 100 cm.

Configure the tag for raw range from zero (Raw Zero Scale) to five (Raw Full Scale). Select **Linear**, and set the engineering range from 50 (Eng Zero Scale) to 100 (Eng Full Scale).

### Example—Square Root Scaling

A flow meter measures the flow rate of a liquid using a differential pressure reading. The device server provides 4–20 mA readings. The actual flow is measured in gallons per minutes (GPM). 4 mA corresponds to 0 GPM; 20 mA corresponds to 100 GPM.

Configure the tag for raw range from 4 (Raw Zero Scale) to 20 (Raw Full Scale). Select **Square Root Scaling** and set the engineering range from 0 (Eng Zero Scale) to 100 (Eng Full Scale).

### Assigning Units to an Analog Tag

Use the **Engineering Unit** field to assign units to a tag. If the desired unit is not in the list, select **Enter New** and enter the desired unit. In the previous example, you select units of GPM.

## Scaling Discrete Tags

The only scaling available for discrete (Boolean) tags is invert scaling.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click a discrete tag to display the **Tag Configuration** dialog box.

3. Click the **Scaling** tab.

4. Place a checkmark in the **Invert Data** checkbox for the Tag Engine to invert the discrete value when it communicates with the device server.

5. Click the **OK** button.

6. Select **File»Save** to save the changes.

## Scaling Bit Array Tags

Bit array tags can have invert and/or select mask scaling.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Double-click a bit array tag to display the **Tag Configuration** dialog box.

3. Click the **Scaling** tab.

4. Place a checkmark in the **Scale Data** checkbox.

5. Set which bits you do not need in the **Scaling Select Mask** textbox.

   Mask scaling applies only to bit arrays. LabVIEW stores a bit array as a number, translating it back into a 32-bit array when needed. The array holds a collection of 32 values, each value limited to 1 or 0. Mask

scaling lets you decide which of these array members to pay attention to. Any bit filtered out by the mask shows up as a zero; bits not filtered out show up as zero or one, reflecting the actual value. A mask of

```
xxxxxxxxxxxxxxxxoooooooooooooooo
```

where x marks a position to be filtered out and o marks a place to report, would read the bit array

```
10110011100011001100111000110010
```

as the following:

```
00000000000000011100111000110010
```

6. Set which bits are inverted between the device server and the Tag Engine in the **Scaling Invert Mask** textbox. You can invert only the bits selected in the **Scaling Select Mask** textbox.

✎  **Note**   The invert mask does not invert bits that are masked out.

7. Click the **OK** button.

8. Select **File»Save** to save the changes.

Table 4-3 shows examples of tags configured for bit array scaling.

**Table 4-3.**  Bit Array Scaling Examples

| Tag Name | Length | Raw Value | Scaling Invert Mask | Scaling Select Mask | Scaled Value |
|----------|--------|-----------|---------------------|---------------------|--------------|
| Tag 1 | 8 | 0x0F | 0x00 | 0xFF | 0x0F |
| Tag 2 | 8 | 0x0F | 0x33 | 0xFF | 0x3C |
| Tag 3 | 8 | 0x0F | 0x33 | 0x0F | 0x0C |
| Tag 4 | 8 | 0x0F | 0x00 | 0x33 | 0x03 |
| Tag 5 | 8 | 0x0F | 0x33 | 0x33 | 0x30 |
| Tag 6 | 16 | 0x0FF0 | 0x000F | 0x00FF | 0x00FF |

# Setting Alarms

Alarms are useful for notifying users of abnormal conditions for a given tag. These attributes include whether to enable alarms, under what circumstances a tag is in alarm, the priority level of an alarm, and how alarms are acknowledged. Each alarm limit has a priority, ranging between 1 and 15. In the LabVIEW DSC module, 15 is the highest priority and 1 is the lowest.

Alarms include two main types:

- Alarms based on status
- Alarms based on tag values

Configuration for alarms based on tag values is specific to data type. Therefore, many alarm attributes apply to only a subset of the tag data types. Refer to Chapter 6, *Alarms and Events*, for more information about accessing and displaying alarms and events.

Complete the following steps to set alarms.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).
2. Double-click a tag to display the **Tag Configuration** dialog box.
3. Click the **Alarms** tab.
4. Place a checkmark in the **Enable Alarms** checkbox.

   Alarms are generated depending on the value or state of a tag. The alarms based on value vary with the tag data type. For any tag, if the status is bad, a bad status alarm is generated. By default, **Bad Status Alarm** is enabled and has the highest priority (15).
5. Set the alarm attributes. The available attributes vary depending on the type of tag you are configuring.
6. Click the **OK** button.
7. Select **File»Save** to save the changes.

## Setting Alarms for Analog Tags

Analog tags have four alarm levels: HI_HI, HI, LO, and LO_LO. By providing separate alarm levels, you can provide more information about the nature of the alarm condition. Alarms are calculated after scaling is performed. Alarm levels are expressed in engineering units.

## Setting Alarm Deadband on Analog Tags

Alarm Deadband defines how much a tag value must change from the alarm limit before it is considered normal. For example, if a tag that represents a temperature value hovers near an alarm limit of 40 degrees Celsius, the tag might go in and out of alarm many times in a relatively short period of time. Table 4-4 shows examples of events with Alarm Deadband set to 0.0%.

**Table 4-4.** Events with Alarm Deadband = 0.0%

| Time | Value | Event | Alarm Type |
|------|-------|-------|------------|
| 9:15:05 | 40.1 | Yes | HI |
| 9:15:10 | 39.9 | Yes | Normal |
| 9:15:15 | 40.1 | Yes | HI |
| 9:15:20 | 38.5 | Yes | Normal |

This type of situation clogs event files with redundant information and can cause operators some frustration in having to acknowledge alarms constantly when the tag has not changed significantly. You can use the Alarm Deadband to alleviate this problem.

For the tag to go into alarm, it must go above the exact Alarm Value (in the previous example, 40). However, to be considered normal again, it must leave the Alarm Value by an amount greater than the Alarm Deadband. For example, if the range is 0 to 100 degrees Celsius, an Alarm Deadband of 1.0% (one degree Celsius) eliminates unnecessary events. Table 4-5 shows examples of events with Alarm Deadband set to 1.0%.

**Table 4-5.** Events with Alarm Deadband = 1.0%

| Time | Value | Event | Alarm Type |
|------|-------|-------|------------|
| 9:15:05 | 40.1 | Yes | HI |
| 9:15:10 | 39.9 | No | HI |
| 9:15:15 | 40.1 | No | HI |
| 9:15:20 | 38.5 | Yes | Normal |

## Setting Alarms for Discrete Tags

Discrete tags have two alarm states—either the tag is in alarm or it is not. You can determine whether a discrete tag is in alarm when it is ON (high) or OFF (low).

## Setting Alarms for Bit Array Tags

You can enable one of two types of alarms for bit array tags. **Alarm on Any** indicates the overall tag is in alarm if any of its bits are in alarm state. **Alarm on All** means the tag is in alarm only if all of the bits are in the alarm state. You can use the Invert Mask to determine the bits that should use

alarm on low (OFF) rather than the default alarm on high (ON). You can use the Select Mask (logical AND) to determine the bits that should be considered for the alarm. If you have bits in the Select Mask that are zero (OFF), these bits are not used in calculation of the tag alarm state.

## Setting Alarms for String Tags

String tags have no alarm states based on tag value. They only support Bad Status alarms.

## Keeping an Alarm Unacknowledged after the Alarm Returns to Normal

On the **Alarms** tab in the **Tag Configuration** dialog box, select the **Alarm Acknowledgement Mode** field and select either **Auto Ack on Normal** or **User Must Ack**:

- **Auto Ack on Normal**—With this option enabled, when a tag returns to normal state, the alarm is automatically acknowledged. A message is logged to the event file if event logging is turned on for the tag. By default, **Auto Ack On Normal** is enabled.

- **User Must Ack**—With this option enabled, an alarm remains unacknowledged until the operator acknowledges the alarm.

# Determining When to Use Memory Tags

Use memory tags when you want to perform alarm calculations or log historical data and event information on data that are either a software-generated values or combinations of values from different I/O tag readings. You do not need to use a memory tag for program variables unless you want to use the historical and event logging or alarm management capabilities of the Tag Engine.

## Scenario 1—Use a Memory Tag

Suppose a simple device server returns several data values, but you are interested in some linear combination of these I/O points. The design of the device and its server software might not allow you to combine these values within the server to make a single tag. But the operators need historical trends and alarm management based on this calculated value. In this situation, you could read individual tag values and calculate the linear combination in the application, then write the calculated value to the

memory tag in the Real-Time Database. The Tag Engine would perform historical logging and alarm calculations according to the memory tag configuration.

## Scenario 2—Do Not Use a Memory Tag

Suppose an HMI calculates and displays the difference between the current temperature reading and a previous temperature reading. Although the individual readings are logged for historical trends, the difference between them is not. If you do not need to perform any alarm management or historical logging based on the difference, you do not need to use a memory tag.

## Creating a Memory Tag

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).
2. Select **Edit»Create** and select the type of tag you want to create.
3. Click the **Connection** tab.
4. Set **Tag Access** to **Memory**.
5. Select any other settings you want for the memory tag.
6. Click the **OK** button.
7. Select **File»Save** to save the changes.

# Customizing the Tag Configuration Editor View

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).
2. Select **Edit»Column Setup** to display the **Select Tag Fields to View** dialog box. You can display the same columns of information, describing every tag attribute, as in the spreadsheet import/export options in the Tag Configuration Editor.
3. Select tag fields in the **Available Tag Fields** listbox and click the **Add** button to move the fields to the **Fields to View** listbox. Click the **All** button to move all fields to the **Fields to View** listbox.
4. Select tag fields in the **Fields to View** listbox and click the **Move Up** or **Move Down** buttons to customize the order of appearance of columns. You also can drag and drop fields to rearrange them.
5. Click the **OK** button.

# Accessing Tags Over a Network

A LabVIEW DSC module *server* is a computer that allows tags configured in the current .scf file to be accessed by other computers connected to it. A *client* is a computer that gets its data through tags from one or more LabVIEW DSC module servers. A LabVIEW DSC module server also can act as a client and get its data from other LabVIEW DSC module server computers.

A .scf file for a LabVIEW DSC module client can contain network tags from multiple LabVIEW DSC module servers, as well as other servers. Refer to the *Importing Network Tags* section earlier in this chapter for more information about importing network tags.

Access to data via DataSocket or across the network is subject to security access rights. Refer to the *Setting Data Access* section in Chapter 8, *Security*, for more information.

## Accessing Tags Over the Network using BridgeVIEW

If you are using BridgeVIEW in your system, you have to perform some additional steps to make sure tags created in the LabVIEW DSC module can be accessed by BridgeVIEW applications. This is also true for BridgeVIEW applications you might be running using the LabVIEW DSC module. If you import a set of tags from a .scf using BridgeVIEW 2.*x* networking, you can only import from one .scf file per computer. Also, you must make sure that the BridgeVIEW Engine running as a data server is using the correct .scf file for your client application to work properly.

1. On the server computer, open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Open the .scf file that contains the tags you want to access over the network.

3. Select **Configure»Network»BridgeVIEW 2.x Networking»Allow Network Access**. Select **Configure»Network»BridgeVIEW 2.x Networking»Clients have write access** if you want network clients to be able to write to the tags in the .scf file.

4. Select **File»Save** to save the changes.

5. On the client computer, open the Tag Configuration Editor.

6. Select **Configure»Network»BridgeVIEW 2.x Networking»Import Network Configuration** to display the **Select Tags for Network Import** dialog box. Only tags exposed through BridgeVIEW 2.*x* networking appear.

7.  Click the **Add** buttons to add the tags you want to import to the **Selected Tags** listbox.

8.  Click the **Import** button.

9.  Select **File»Save** to save the changes.

## Sharing Tag Data with Other Applications

You can access the real-time value of tags from other applications. You can use DataSocket to access tag data, or you can use any OPC client to access the LabVIEW DSC module. The Tag Engine can operate as an OPC server. When you access data from the Tag Engine acting as an OPC server, client software such as Lookout or some other OPC client software sees the Tag Engine as an OPC server named `National Instruments.OPCLabVIEW`. The tag names you set in the `.scf` file appear as the OPC item names.

Because the Tag Engine is an OPC server, an OPC client can launch it. When an OPC client launches the Tag Engine, the Tag Engine loads the current `.scf` file and any servers that the `.scf` file tags require. An OPC client cannot specify which `.scf` file the Tag Engine should use. The Tag Engine can use only the current `.scf` file, so you must make sure the correct `.scf` file is set as the default if you want this type of access.

If the `.scf` file contains memory tags that hold values calculated by one of your VIs, those values do not update unless you run the LabVIEW VI manually. However, neither the Tag Engine nor an OPC client run LabVIEW, so you must launch the correct VI manually for the memory tag data to update.

# Viewing Tag Engine Status

Launch the Tag Engine and open the Engine Manager. If the Tag Engine is already launched and running, the Engine Manager might be minimized and appear only as an icon in the system tray of your Windows taskbar. Double-click the Tag Engine icon to open the Engine Manager display.
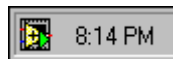


Otherwise, you can launch the Engine Manager when the Tag Engine is running by selecting **Tools»Datalogging & Supervisory Control»Launch Engine»Show**.

You can leave the Engine Manager display minimized unless you want to use it to start or stop the Tag Engine, start or stop historical logging, event logging and printing, view system events, or view server information.

The Tag Engine works with tags configured in the `.scf` configuration file and created in the Tag Configuration Editor. The Engine Manager shows the current state of the Tag Engine.
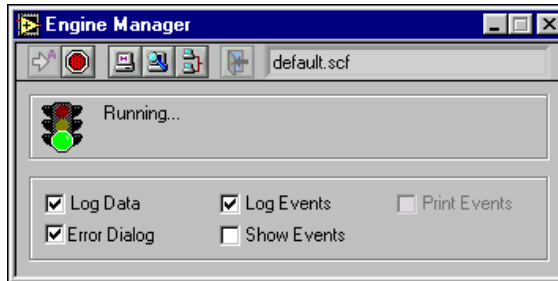


Table 4-6 describes the fields in the **Engine Manager** window.

**Table 4-6.** Engine Manager Field Descriptions

| Field | Description |
|---|---|
| Engine Status (next to the traffic signal) | Displays the current status of the Tag Engine—whether launching, running, or stopped. |
| Log Data | Turns on or off logging of historical data to file. To turn on logging of historical data to file automatically, select **Configure»Historical** from the Tag Configuration Editor, then select **Start logging on system start-up**. If you do not have a valid historical log path configured, logging of historical data to file is disabled. |
| Log Events | Turns on or off logging of alarms and events to file. To turn on logging of alarms and events to file automatically, select **Configure»Events** from the Tag Configuration Editor, then select **Start event logging on system start-up** on the **Event Logging** tab. If you do not have a valid event log path configured, logging of alarms and events to file is disabled. |
| Print Events | Turns on or off printing of alarms and events to a line printer. To turn on printing of alarms and events automatically, select **Configure»Events** from the Tag Configuration Editor, then select **Start printing on system start-up** on the **Printing** tab. If you do not have a printer configured, printing of alarms and events is disabled. |

**Table 4-6.** Engine Manager Field Descriptions (Continued)

| Field | Description |
|---|---|
| Error Dialog | Enables or disables showing the **Error** dialog box. If this checkbox contains a checkmark, a **System Error Display** dialog box appears for you to acknowledge the event when a system error occurs. |
| Show Events | Shows or hides the System Event Display. The System Event Display shows the following when **Show Events** contains a checkmark: <br>• LabVIEW system events <br>• When the Tag Engine started and stopped <br>• Which servers have been launched <br>• Any system errors that have occurred |
| **Toolbar** | |
| Start the Tag Engine | Starts the Tag Engine. |
| Stop the Tag Engine | Stops the Tag Engine and shuts down any running servers. |
| Start Tag Monitor | Opens the Tag Monitor. |
| View Servers in Use | Displays the servers in use by the Tag Engine. |
| View Client Connections | Shows all the computers currently accessing data from the LabVIEW application. |
| Exit the Tag Engine | Closes and exits the Tag Engine application. |

# Configuring Tag Engine Parameters

The Tag Engine has several default settings for parameters. Complete the following steps to override these defaults.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Select **Configure»Engine**.

3.  Select among the Tag Engine options on the various tabs.

✎ **Note**  Although you can configure these parameters, it is highly recommended you maintain the default values. If you use a large number of string tags and the string tags are large or change rapidly, you might need to increase the input queue binary size to be larger than the default 2,000 bytes.

4.  Click the **OK** button.

The Tag Engine allocates certain amounts of memory for various queues. You can configure some of the parameters used by the Tag Engine and Tags VIs to allocate memory for the Tag Engine buffers.

# Monitoring and Writing Tag Values

You can use the Tag Monitor to monitor the value, timestamp, alarm state, and quality for selected tags in the system, as well as write the value to an output or input/output tag.

1.  Open the Tag Monitor (**Tools»Datalogging & Supervisory Control»Monitor Tags**).

2.  Navigate to the tags you want to monitor using the tree in the left pane. Select **View»Default** to restore the full tree.

    You can see tags both directly in your local computer, under the **My Computer** node, and across the network, under the computer name under the **Network** node. You also can see data from other software and devices on other networked computers. To find tags or data on another computer, you must first register the computer. Refer to the *Registering Networked Computers* section of Chapter 9, *Networking and Deploying Applications*, for more information about registering and unregistering computers.

3.  To select a tag for monitoring, double-click the tag to move it to the tag display pane on the right. You also can select one or more tags and drag them to the tag display pane or you can select tags, right-click, and select **Add** from the shortcut menu. The **Quality** column shows status information for the tags.

4.  To add, remove, write to, or edit the properties of a tag, select the tag and select the corresponding options in the **Items** menu. You also can right-click a tag to access these options.

5.  Select **View»Refresh** to refresh the tree view in the left pane and the alarm view in the bottom right pane. The tags in the tag display pane update continuously and do not need to be refreshed.

6.  Select **File»Save As** to save different sets of tags to monitor.

**5**

# Displaying and Controlling Data

An HMI is the interface through which an *operator* interacts with the LabVIEW system and with the outside environment that LabVIEW monitors and controls. In LabVIEW, you use VIs as your HMI. The LabVIEW Datalogging and Supervisory Control (DSC) module installs custom controls, indicators, VIs and functions to make HMI creation easier. Refer to Chapter 1, *Introduction*, for more information about these objects.

The example block diagrams shown in this chapter are taken from some of the examples you can find in `labview\examples\lvdsc`.

The LabVIEW DSC module includes a set of VIs with which you can control your HMI, access the Real-Time Database and the Citadel historical database, perform calculations and logic, and change between different displays. The LabVIEW DSC module VI library adds Alarm and Event VIs, Historical Data VIs, System VIs, Tags VIs, and Tag Attributes VIs.

There are several general LabVIEW programming principles with which you should be familiar before you build an HMI. For more advanced HMI programming, you also should know how to use Property Nodes and the VI Server. Refer to the LabVIEW documentation for more information about Property Nodes and the VI Server.

To develop an HMI application, create the front panel interface and then complete the block diagram.

## Building Front Panels

Refer to the LabVIEW documentation for more information about building front panels. This section describes the specific front panel features you get with the LabVIEW DSC module.

A front panel is the user interface of a VI. You might want to divide an HMI into several front panels so the operator can navigate through them using buttons on the screen. The Panel Wizard can help you do this by automatically generating code and attaching it to front panel buttons.

Use the HMI Wizard to connect controls and indicators to tags. You invoke the HMI Wizard on the front panel, but it adds elements to the block diagram.

Graphics, both as decorations and as controls and indicators, can make front panels easier to operate and more informative.

# Creating Front Panel Controls and Indicators

You build the front panel of a VI with a combination of controls and indicators representing the values of the tags. Controls supply data, and indicators display data. Select controls and indicators on the **Controls** palette and place them on the front panel.

## Connecting Controls and Indicators to Tags

1. Place the control or indicator on the front panel.

2. Right-click a control or indicator and select **HMI Wizard** from the shortcut menu.

3. Select the tag to which you want to connect in the **Tag** pull-down menu. Click the **Create New Tag** button, shown at left, to create a new tag.

4. Select among the following options on the **Data Connection** tab:

    • **Match control label to tag name**—Matches the label on the control or indicator to the tag name.

    • **Match scale to tag configuration**—Adjusts the object scale to match the maximum and minimum of the tag you select.

5. Click the **Alarms** tab.

6. Select among the following options on the **Alarms** tab:

    • **Change colors with alarm state**—Changes the color of the control or indicator when the tag is in an alarm state.

    • **Blink when**—Makes the control or indicator blink when the tag is in an alarm state or when the alarm is unacknowledged.

7. Click the **OK** button to generate any necessary block diagram code for the options you selected.

Refer to the *Auto-Generating Code with the HMI Wizard* section for more information about working with the HMI Wizard and the block diagram elements it generates.

## Creating a Tag Control

The tag control lets you select a tag name from the list of all configured tags in the active `.scf` file. You also can enter the name of the tag you want through the keyboard.

The tag control is located on the **Controls»I/O** palette. You also can use a tag control in an array.

If you organized tags into groups, you see a folder icon for each tag group. You can select a tag group as well as an individual tag. You can use tag groups to acknowledge all alarms on all tags in a group. You cannot use a tag group to read or write all tags in the group.

By default, the tag control does not allow you to enter a name that is not contained in the current `.scf` file. But you can configure each tag control, indicator, or constant to permit entry of names not contained in the current `.scf` file by right-clicking the tag object and selecting **Allow Undefined Tags** from the shortcut menu.

You can create, edit or copy a tag by right-clicking the tag control, indicator, or constant, and selecting **Create Tag**, **Edit Tag**, or **Copy Tag** from the shortcut menu. Selecting any of these options invokes the Tag Wizard. Any new or changed tags are automatically saved to the active `.scf` file, and the list of available tags is then updated. Refer to the *Configuring Tag Attributes* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information about configuring tags.

Tag controls and indicators saved with default values and tag constants on the block diagram retain the tag name or group selected when you save the VI. The name contained in the tag control, indicator, or constant is dimmed if the name is not contained in the currently selected `.scf` file. This might happen if the tag name was deleted from the `.scf` file or the VI was created using a different `.scf` file than the one currently loaded. If you try to run the VI in this condition, you get a system error for each tag that is undefined in the current `.scf` file.

## Adding and Customizing Graphics

The **Controls** palette includes Boolean controls and indicators such as pipes, valves, and pumps, and numeric controls and indicators such as tanks and hoppers. These images change to reflect the state of the tags you connect them to.

The LabVIEW DSC module also includes an extensive library of supervisory control and other images ranging from basic geometric shapes to drawings of special equipment. Use the Image Navigator to access these images.

Select **Tools»Datalogging & Supervisory Control»Image Navigator** to open the Image Navigator. Refer to the *Image Navigator Help* for more information about editing, copying, and importing and exporting graphics, and other Image Navigator options.

# Building an Application with Multiple Front Panels Using the Panel Wizard

If size or other considerations make it desirable, consider dividing the HMI into several front panels so the operator can navigate through them using onscreen buttons.

Use the Panel Wizard to generate a front panel navigation system for operators. With the Panel Wizard, you can attach code to buttons that, when clicked by the operator, open VIs. The Panel Wizard only operates on Boolean controls. If you are unfamiliar with LabVIEW, the Panel Wizard can be a quick way to produce applications that can navigate among multiple windows and front panels.

1.  Save the VI. Because the code created by the Panel Wizard contains file path information, some features do not generate correct block diagram code until the calling front panel is saved to disk.

2.  Place a button on the front panel.

3.  Right-click the button and select **Panel Wizard** from the shortcut menu.

4.  In the **With this Control** pull-down menu, select whether to open or close a front panel when the Boolean control is activated. The Panel Wizard can create block diagram code to open several types of front panels:

    •   Login prompt.

    •   Historical Trend Viewer.

    •   Tag Monitor.

    •   VIs that you have created and saved to disk. When opening VIs that you have created, specify the path to the VI by typing it in the **Path to VI** field or by selecting the **Browse** button. You can configure the Panel Wizard to store the file path as a relative

path or as an absolute path. If the VI that you connect to the control is generally kept in a path that is relative to the top-level VI, select the **Relative path** setting.

5.  Click the **Button** tab.

6.  Place a checkmark in the **Match button label to action** checkbox to change the button label to match the VI name you are using the button to call. If you are using the wizard to invoke the login dialog, Historical Trend Viewer, or Tag Monitor, the label reflects those choices. Placing a checkmark in the **Match Boolean text to action** checkbox does the same for the Boolean text.

7.  Select a **Mechanical Action** option. These options function the same way as the mechanical action options available by right-clicking a Boolean control.

8.  Click the **Polling** tab.

9.  Select a **Poll this control** option.

    *   **Until engine shutdown**—If you run the calling VI while the Tag Engine is not running, the VI polls when started and continues to poll until the Tag Engine starts and then shuts down. However, if the Tag Engine is restarted after having stopped, the VI does not resume polling until you restart the calling VI.

    *   **Always**—Always polls the control.

# Block Diagram Tools and Functions

Refer to the LabVIEW documentation for more information about building block diagrams. This section describes the block diagram features specific to the LabVIEW DSC module.

## Auto-Generating Code with the HMI Wizard

Use the HMI Wizard to generate common sections of block diagram code. To open the HMI Wizard, right-click a front panel object and select **HMI Wizard** from the shortcut menu. The HMI Wizard operations vary, depending on the front panel object. The HMI Wizard makes a DataSocket connection between a front panel control or indicator and a tag, and, when necessary, generates a subdiagram for any configuration that you specify. Refer to the LabVIEW documentation for more information about DataSocket.

Refer to the *Connecting Controls and Indicators to Tags* section for more information about generating block diagram code from a front panel object.

Table 5-1 lists front panel objects and describes how the HMI Wizard operates on each of them.

**Table 5-1.** HMI Wizard Operations for Various Front Panel Objects

| HMI Function | Front Panel Object | Description |
|---|---|---|
| Control analog tags | Numeric control | Invoke the HMI Wizard on a numeric control to associate an analog output tag value with that control. You can set the colors and blink options on the **Alarms** tab. |
| Display analog values | Numeric indicator | Invoke the HMI Wizard on a numeric indicator to associate an analog input tag value with that indicator. You can set the color and blink options on the **Alarms** tab. |
| Control discrete tags<br><br>Acknowledge alarms | Boolean control | Invoke the HMI Wizard on a Boolean control to associate a discrete output tag value or an alarm acknowledgement action with that control. To change the control association from tag value to alarm acknowledgement mode, change the **Attach Control to** field from **Tag Value** to **Alarm Acknowledgement**. Select a set of tags that requires acknowledgement when the control is TRUE. |
| Display discrete values<br><br>Indicate an alarm state | Boolean indicator | Invoke the HMI Wizard on a Boolean indicator to associate a discrete input tag value or an alarm state with that indicator. To change the control association from tag value to alarm state, change the **Attach Indicator to** field from **Tag Value** to **Tag Alarm State**. Select the tag for which the indicator displays the alarm state. |
| Control string tags | String control | Invoke the HMI Wizard on a string control to associate a string output tag value with that control. You can set the colors and blink options on the **Alarms** tab, and specify the updates to happen only when the control value changes. |

**Table 5-1.** HMI Wizard Operations for Various Front Panel Objects (Continued)

| HMI Function | Front Panel Object | Description |
|---|---|---|
| Display string values | String indicator | Invoke the HMI Wizard on a string indicator to associate a string input tag value with that indicator. You can set the color and blink options on the **Alarms** tab. You can display strings in normal form or as hex codes. You also can set the indicator to use a scrollbar if you anticipate any need to display long strings. |
| Display alarm summary | Alarm summary display listbox or any table indicator | Use the alarm & event display control on the **Controls»Alarms and Events** palette. This ActiveX control connects directly to the Citadel historical database and does not need to be connected or configured with the HMI Wizard.<br><br>You can set the HMI Wizard to Alarm Summary mode by setting the value of the **Use this indicator for** field to **Alarm Summary**. Select a set of tags whose alarms require monitoring. You also can set colors of acknowledged and unacknowledged alarms and column format of the summary. Preformatted alarm summary indicators are located on the **Controls»Alarms and Events** palette. |
| Display event history information | Event history display listbox or any table indicator | You also can invoke the HMI Wizard on a multicolumn listbox or table indicator to obtain a history of past events and alarms (Event History). For an event history display, you can set the HMI Wizard to Event History mode by setting the value of the **Use this indicator for** field to **Event History**. Select a set of tags whose history needs to be displayed. You also can set colors of acknowledged and unacknowledged alarms, normal and event entries, and column format of the summary. You can set filters to determine how many and what sort of alarms are displayed.<br><br>Preformatted event history indicators are located on the **Controls»Alarms and Events** palette. |

**Table 5-1.**  HMI Wizard Operations for Various Front Panel Objects (Continued)

| HMI Function | Front Panel Object | Description |
|---|---|---|
| Display a real-time trend | Real-time trend or waveform chart indicator | Invoke the HMI Wizard on a real-time trend or waveform chart indicator to select a set of tags for which the values need to be displayed in a chart (real-time trend). You also can set engineering units for the display, percentage of scale, and the time (in seconds) between updates. |
| Display a historical trend | Historical trend or XY graph indicator | Invoke the HMI Wizard on a historical trend or XY graph indicator to select a set of tags for which the values need to be displayed in an XY graph (historical trend). |

# Changing Auto-Generated Block Diagram Code

The lock in the upper-right corner of a block diagram structure indicates that the subdiagram has been locked.

While the subdiagram is locked, you can right-click the front panel object, select **HMI Wizard** from the shortcut menu, and change your selections in the dialog box.

To edit the block diagram directly, right-click the subdiagram and select **Release Wizard Lock** from the shortcut menu. After you release the lock, any wizard association is broken. You can no longer use the HMI Wizard to make changes.

# Tag Data Types

Many LabVIEW DSC module VIs and functions require a tag name as an input. So in addition to standard LabVIEW data types, the LabVIEW DSC module includes the tag data type. All LabVIEW DSC module functions that can operate on tags or tag groups use the tag data type. The tag data type is marked with a valve icon. Constants and wires on the block diagram carrying this tag information are displayed in purple. Tag constants also can be contained in arrays. Many LabVIEW DSC module VIs operate on arrays of tags.

The tag constant used as constant input for a tag and/or the tag attribute VI is located on the **Functions»Tags** palette. The **Functions»Tags** palette also contains functions that convert between a tag data type and a string

data type, and a special **Not a Tag** constant. The **Not a Tag** function is a constant that reports Not a Tag. This constant is most often used in case structures when one condition contains a tag name and another does not.

The tag data type uses tag and tag group names from the tag configuration file (`.scf`). When you start LabVIEW, the tag data type list of available tag names and tag group names is automatically updated from the default `.scf` file. The default `.scf` file is the last file you edited in the Tag Configuration Editor. If the list of names is empty, you need to open an `.scf` file in the Tag Configuration Editor.

# Tasks Common to Both Tags and Alarms & Events VIs

Use the VIs and functions located on the **Functions»Tags** and **Functions»Alarms & Events** palettes to operate on tags by wiring the tag name or tag group name into the **tag name** or **group/tag name** input of the VI. Some VIs accept arrays of tag names or tag and tag group names.

## Reading Tag Engine Status Programmatically

The Tags VIs and Alarms and Events VIs return several flags that indicate the state of the Tag Engine. They return a Boolean **error** flag to indicate whether the operation was successful. If the **error** flag is TRUE, the tag-specific information returned by the VI might not be valid. Some VIs also return a more detailed **value status** variable.

When an application runs one of the Tags VIs or Alarms and Events VIs, that VI checks the status of the Tag Engine. If it is not running, the VI automatically starts the Tag Engine, which loads and executes all required device servers. When the VI returns, the Tag Engine remains running until you shut it down either programmatically or through the Engine Manager. LabVIEW DSC module VIs that do not access the Real-Time Database, such as the Historical Data VIs, can run without the Tag Engine running and do not launch the Tag Engine.

You can monitor the shutdown status of the Tag Engine with the **shutdown** output of any of the Tags VIs or Alarms and Events VIs, or with the Get Engine Status VI. If TRUE, this output indicates that the Tag Engine is in the shutdown state, and the application must finish running so that shutdown can finish. If the Tag Engine goes into the shutdown state while these VIs are waiting on an event, the VI terminates the wait and returns immediately to the calling block diagram. You can use this output to tell the block diagram to complete execution.

## Starting or Stopping the Tag Engine Manually

You can start or stop the Tag Engine manually in the Engine Manager, the Tag Configuration Editor, or the **Tag Utilities** toolbar. As the Tag Engine shuts down, first it checks whether any application is running that requires its services. If so, it waits until that application halts before shutting down. After it shuts down, it sends a shutdown message to the device servers.

## Setting the Database Read Rate

All VIs that read information from the LabVIEW DSC module Real-Time Database can return information immediately or wait for the database to be updated with new information before returning. The **timeout** input controls this behavior. This input tells the VI how long to wait, in seconds, for the tag information to be updated in the Real-Time Database. Table 5-2 describes the behavior for each **timeout** value.

**Table 5-2.** Timeout Values and Behavior

| VI Timeout Input | Behavior |
|---|---|
| 0 | VI immediately reads the database and returns the current tag information. |
| Less than 0 | VI continues to wait until the tag is updated or the Tag Engine shuts down. |
| Greater than 0 | VI waits until the tag is updated in the database, or the timeout period is exceeded, whichever occurs first, then reads the database and returns the current tag information. |

If you wire nothing to the **timeout** input of the block diagram, the default value is 0—the VI reads the database and returns immediately. How you use the **timeout** input depends on whether you want to implement event-driven or polled programming techniques in the HMI.

## Updating Tag Displays When Values Change

You can use a **changed?** output to update the front panel only if information has changed. All VIs that read information from the LabVIEW DSC module Real-Time Database have a **changed?** output that is TRUE if the returned information is new or updated. If the VI returns and **changed?** is FALSE, the VI might have timed out, or the information in the database did not change since the last read.

### Determining When the HMI Needs to Be Initialized

Some of the more advanced Tags VIs and Alarms & Events VIs also return an **initialize headers** or **config changed** output that tells the VI whether the front panel object needs to be initialized with new information. In most cases, this corresponds to the first time the VI is called, and you only need to update that part of the user interface once.

# Controlling System Behavior Programmatically

As you develop more sophisticated user interfaces, you might find that you need to exercise control over portions of the LabVIEW environment from your own applications. Use the VIs located on the **Functions»System** palette to control the Tag Engine programmatically.

The System VIs provide you with several options for controlling actions such as launching and shutting down the Tag Engine, and enabling and disabling logging.

📝 **Note**   One of these functions, the Get Engine Status VI, is what the HMI Wizard looks for when deciding whether to insert a Tag Engine monitoring loop into the block diagram. If you use this VI before using the HMI Wizard, you might want to create a Tag Engine monitoring loop yourself.

## Starting or Stopping the Tag Engine Programmatically

Use the Engine Launch VI to launch the Tag Engine programmatically with a specified configuration file. Use the Engine Shutdown VI to stop the Tag Engine and the servers currently executing.

## Starting or Stopping Historical Logging Programmatically

Use the Enable Historical Data Logging VI to start historical logging. If the input value is TRUE, historical logging is turned on if it is currently off. If the input value is FALSE, historical logging is turned off if it is currently on.

## Starting or Stopping Event Logging Programmatically

Use the Enable Event Logging VI to start logging events, including alarms. If the input value is TRUE, event logging is turned on if it is currently off. If the input value is FALSE, event logging is turned off if it is currently on.

## Starting or Stopping Event Printing Programmatically

Use the Enable Printing VI to start printing alarms and events. If the input value is TRUE, event printing is turned on if it is currently off. If the input value is FALSE, event printing is turned off if it is currently on.

# Changing Tag Attributes Programatically

Use the VIs located on the **Functions»Tag Attributes** palette to read or change configuration information about tags programmatically. Most of these *tag attributes* are parameters you can configure for a tag with the Tag Configuration Editor.

While you can change tag attributes programmatically, not all tag attributes can be changed dynamically. To change a tag attribute dynamically means to change it while the Tag Engine is running. You can change all operations, alarms, and most scaling, and raw or engineering range information dynamically. These are the same attributes you can update with the Tag Configuration Editor without shutting down and restarting the Tag Engine.

Other tag attribute changes might require the Tag Engine to be stopped and restarted to take effect.

Tag attribute changes only affect the current Tag Engine process until they are subsequently changed or the Tag Engine stops. If you stop the Tag Engine and start it again, the changes are lost.

There are certain attributes you cannot change dynamically. These attributes require you to edit the .scf file with the Tag Configuration Editor. They include information such as:

• Tag name

• Tag description

• Scaling type

• Engineering unit

• Data type (analog, discrete, bit array, string)

• Tag group name

• Access rights (input only, output only, Input/Output, memory)

• Tag connection information (server, I/O Group, and item)

You can programmatically take a tag on or off scan. If a tag is off scan, it is not processed or updated in the Real-Time Database, alarms are not calculated, and data are not logged. You can start these activities by putting that tag back on scan.

If the Tag Engine is running, you can change tag attributes programmatically with the Set Tag Attribute VI, Set Multiple Tag Attributes VI, or the Tag Configuration Editor. These VIs return an error if the Tag Engine is not running. Use these VIs to change attributes of a tag dynamically, as with logging, alarm, or scaling information, or taking a tag on or off scan.

✎ **Note**  If you programmatically change a tag attribute, the LabVIEW DSC module does not save the change in the .scf file, so you cannot access the changed attribute after your application closes.

Refer to the *Configuring Tag Attributes* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information about tag attributes.

## Example of Changing Tag Attributes Programmatically

The following figure shows a simple case where the scale range for all the front panel level and temperature indicators are set to the engineering scale that is configured for the tag. A unit string display for each **L1** and **Temp** indicator is initialized to the engineering unit for the respective tag. The Get Tag Range and Units VI returns the engineering scale range information in a form that can be wired directly to a control or indicator scale range Property Node. The VI also returns the engineering units configured for the tag.
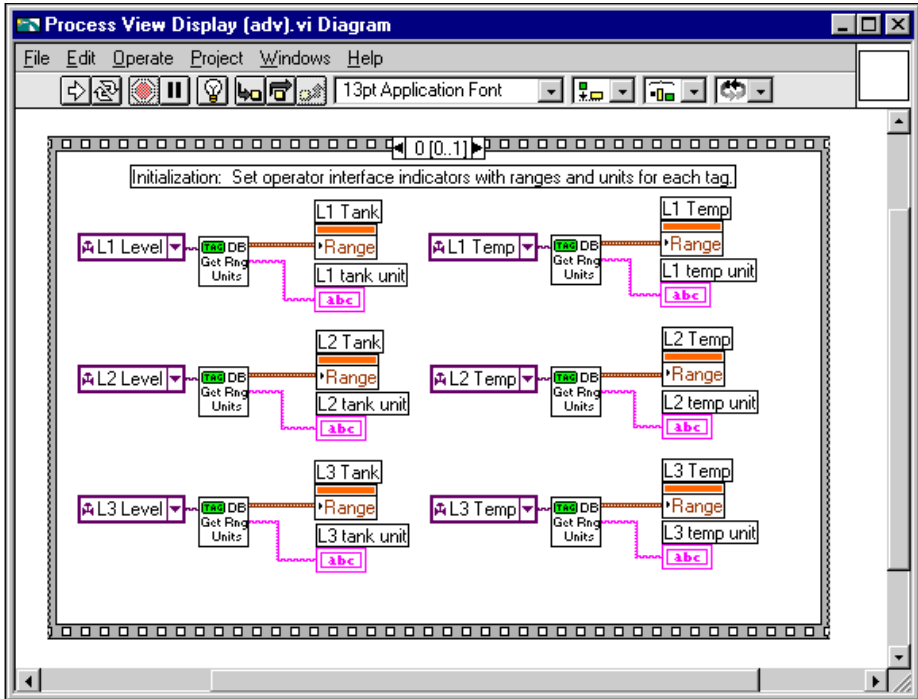
The following figure illustrates the subsequent frame of the Sequence structure. The HMI runs in a loop, monitoring the various tags and front panel controls until the Tag Engine shuts down.
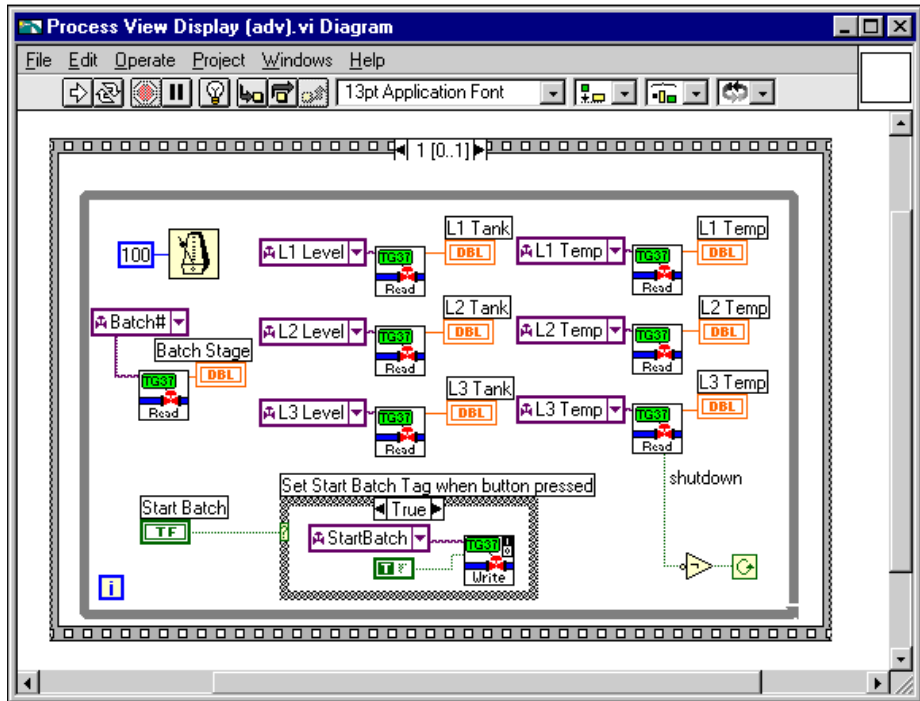
# Setting Timing Execution of While Loops

Normally, you use one or more While Loops on the block diagram, with a single wait operation inside of each loop. Each While Loop executes once after its wait operation completes. The wait operation can be either a polling or event-driven construction. Either technique is appropriate, depending on whether you want regular updates or only need to update following certain events. The following sections describe the different While Loop constructions.

You can wait on multiple events for which timing is not related to each other in parallel on the same block diagram, but you must wait for each event in a separate While Loop.

# Implementing Event-Driven Programming

*Event-driven programming* means the block diagram waits for one or more events to happen and, as each event occurs, the part of the VI waiting on that event is executed. You can develop applications that wait on different events and do operations in parallel by using multiple While Loops in the block diagram.

To implement an event-driven loop, you can use a Tags VIs or Alarms & Events VIs with the **timeout** input wired to a non-zero value.

The following figure shows an example using event-driven programming to monitor tag value and tag alarm state. One loop monitors the value of the **Mixer** tag and another loop monitors alarm information for the **Mixer** tag. These two loops run independently. When the **Mixer** tag value changes, or when 1.00 second has elapsed, the Read Tag VI returns and updates the **Mixer in Alarm**, **Mixer**, **value timestamp**, and **bad value** indicators. When the **alarm state** of the **Mixer** tag changes, or 5.00 seconds have elapsed, the Read Tag Alarm VI returns and updates the **alarm state** indicator, and controls the blinking of the **Mixer in Alarm** indicator. Both loops run in parallel until **shutdown** is TRUE.
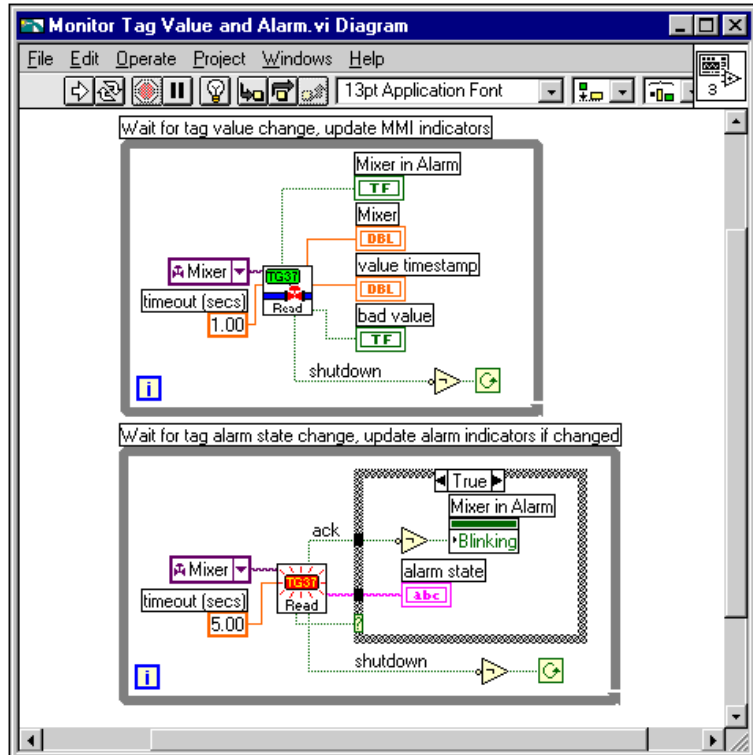
The **Mixer in Alarm Blinking** attribute and the **alarm state** indicator are updated only when the **changed?** output of the Read Tag Alarm VI is TRUE. This example demonstrates how you might use the **changed?** output. In this example, it is not important to use the Case structure because the indicators update only when the displayed information changes.

If you use a large number of indicators or Property Nodes or more complex indicators such as tables and graphs, updating the indicator when changed by using a Case structure in the block diagram can improve the display performance of the VI.

## Implementing Polled Programming

In a polled technique, the wait operation you use inside the While Loop is one of the Time and Dialog functions such as the Wait Until Next ms Multiple function. With this method, the block diagram controls loop execution.

You do not have to use a separate loop for each Tags or Alarms and Events VI. This can be cumbersome to build for a large number of tag reads. The alternative is to poll the database for several tags at regularly timed intervals. You usually need one While Loop in the block diagram to poll the front panel controls so that you can monitor what the operator is doing. Using polling, you can both monitor front panel controls and read in tag values and alarm states.

The following figure shows an example implementing a complex user interface that polls all input tags and the front panel **Start Batch** button at 100 ms intervals. When you leave the **timeout** input unwired, all Read Tag VIs read the LabVIEW DSC module Real-Time Database immediately by default. In this case, you must explicitly add the loop wait time by using the Wait Until Next ms Multiple VI. If you do not, the loop operates as often as possible and requires most of the CPU time.

This example also illustrates use of the Write Tag VIs. In this case, the Write Tag (discrete) VI is called only when the front panel button is clicked. In other cases, you might want to write the tag value at each iteration. You also can use the Write Tag on Change VI to update the RTDB only when the value of the front panel control changes. This can improve over all application performance.
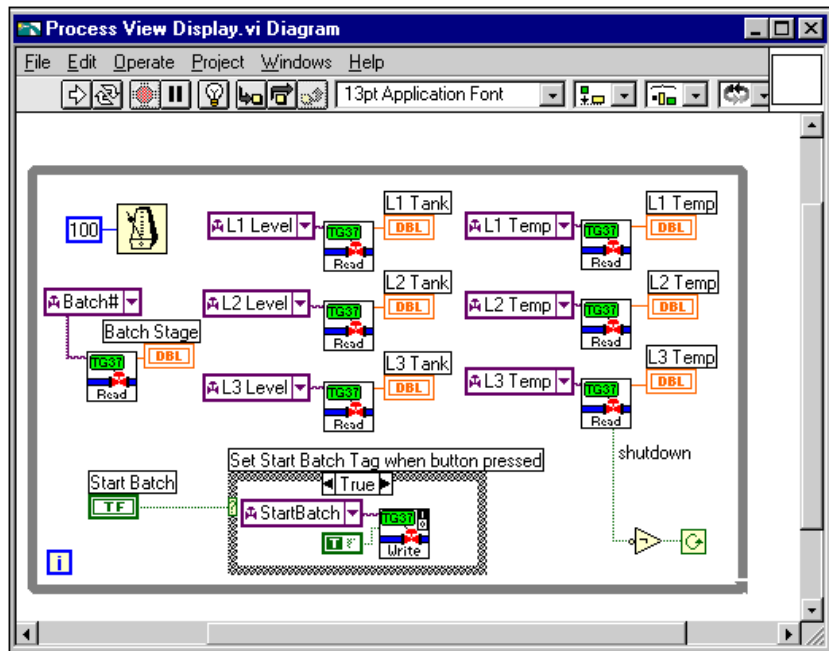
## Initialize and Shut Down Multiple-Loop Applications

When you have a multiple-loop application, you can add initialization code before executing the loops, and some cleanup or shutdown code after all the loops finish executing. You can use the Sequence structure for this purpose. Put the initialization code in the first frame of the Sequence structure, put all the loops in the second frame of the Sequence structure, and put the shutdown code in the final sequence. This guarantees that none of the loops start execution until the initialization code is complete and that all loops complete execution before you execute the shutdown code.

You also can use dataflow programming to enforce sequential operation. In some cases, the block diagram might be easier to read using this technique. It is possible that you might have some data flow between the initialization code and the loops anyway. There is no difference in performance using either technique.

# Displaying Trends

A *trend* is a graph display of tag values over time. With the LabVIEW DSC module, you have several options for displaying trends for real-time and historical data, as described in the following sections.

## Displaying Real-Time Trends

A real-time trend is a display of tag values as they are collected in real time over a relatively short period of time.

1. Select a Real-Time Trend indicator on the **Controls»Trends** palette and place it on the front panel.

2. Build the block diagram in one of the following ways:

   - Generate code automatically by right-clicking the indicator on the front panel and selecting **HMI Wizard** from the shortcut menu. Refer to the *Connecting Controls and Indicators to Tags* section for more information about generating block diagram code from a front panel object.

   - You can assemble the code manually on the block diagram using a While Loop and the Trend Tags VI. Wire the **trend data** output of the Trend Tags VI to the real-time trend indicator terminal.

     Because the Trend Tags VI waits the specified **time interval**, place the VI in its own While Loop to control the loop execution rate. To run other VIs at the same rate that the real-time trend updates, place them in the same While Loop.

The Real-Time Trend indicator updates with a value for each tag
every time the indicator is written to. If a VI using this indicator
runs several times, it still has previous data displayed. For this
reason, you might want to initialize the Real-Time Trend indicator
before the loop begins execution.

You also can control attributes of the Real-Time Trend indicator
such as time scale. The following block diagram shows a single
real-time trend display VI that initializes the time scale of the
trend indicator to the current time using the Get Date/Time In
Seconds function and initializes the interval corresponding to the
Trend Tags VI **time interval** input. It also clears the trend display
by writing an empty array to the **Trends History Data** attribute.
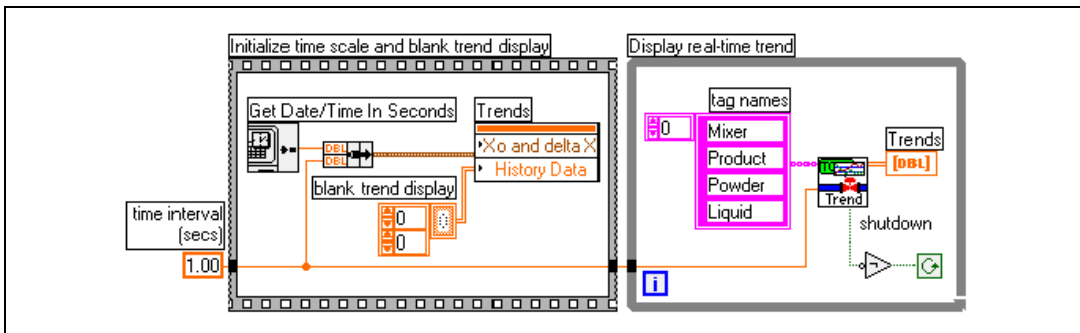


This example illustrates the use of dataflow programming. By wiring the
**time interval** constant through the Sequence structure and into the While
Loop, the While Loop does not execute until the Sequence structure has
completed execution. You also can enforce this order of execution by
placing the While Loop inside the second frame of the Sequence structure.

## Displaying Historical Trends

A historical trend is a display of tag values that have been logged to disk,
usually over a relatively long period of time. If your application is set up to
log values, there are several ways you can display a historical trend. Refer
to the *Extracting and Viewing Historical Data* section of Chapter 7,
*Historical Data Logging and Extraction*, for more information about
displaying historical data.

# 6

# Alarms and Events

This chapter describes how to report, log, and respond to *alarms* and *events* with LabVIEW Datalogging and Supervisory Control (DSC) module applications.

An event is something that happens within the LabVIEW DSC module system. Events can be divided into two groups: *tag events* that pertain to individual tags, and *system events* that pertain to the overall LabVIEW DSC module system. An example of a tag event is a change of alarm state for a tag. Examples of system events include a user logging on, the Tag Engine starting up, or historical logging being turned on.

In the LabVIEW DSC module, an alarm is a specific kind of event related to the value of a tag. An event could be virtually any instantaneous activity such as clicking a mouse button, but an alarm typically has the following characteristics:

• Denotes an abnormal condition

• Occurs under certain, specific conditions

• Must be acknowledged by the user or configured for automatic acknowledgment

Because alarms are generated by tag values, you set most alarm attributes as a part of configuring tags. Refer to the *Setting Alarms* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information. You also enable tag event logging when you configure tags. Refer to the *Configuring a Tag to Log Data or Events* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information.

For the purposes of logging and retrieval, events and alarms are combined.

# Logging and Printing for Alarms and Events

To print logged alarms and events, use the HIST Alarm & Event Query To Spreadsheet File VI on the Alarms & Events VIs palette. You can configure automatic logging and printing for alarms and events in the Tag Configuration Editor as follows.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select **Configure»Events** to display the **Event Configuration** dialog box.

3. Select among the logging and printing options. To open the context help window and view descriptions of these options, press <Ctrl-H> and move the cursor over any field.

**Note**   If you log data to directories created in a secure file system, such as NTFS, you must grant the System account Change or Full Control permissions to the directory. If you do not grant the System account appropriate access to the database directory, Citadel will be unable to create and modify the database files it uses to store historical data and alarms.

4. Click the **OK** button.

5. Select **File»Save** to save the changes.

**Note**   To log remote operator changes of a control as an event, select **Configure»Engine** in the Tag Configuration Editor. On the **Events** tab, place a checkmark in the **Generate Event when Remote User Changes Value** checkbox.

# Viewing Alarms and Events

You can use several different approaches to display and manage alarms and events generated in LabVIEW DSC module applications. Some approaches operate through traditional VIs; others use capabilities built in to National Instruments networking. The multiplicity of approaches is provided not only for flexibility, but also for compatibility with existing BridgeVIEW applications.

• The easiest way to view and acknowledge alarms and events is to use the alarm & event display control located on the **Controls»Alarms and Events** palette. Refer to the following section, *Viewing Alarms and Events with the Alarm & Event Display Control*, for more information.

- You can view alarms and events with an alarm summary display or event history display located on the **Controls»Alarms and Events** palette. Refer to the sections *Using an Alarm Summary Display* and *Using an Event History Display* in this chapter for more information.

- You can use the Tag Monitor to view alarms and events. Refer to the *Monitoring and Writing Tag Values* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW* for more information about the Tag Monitor.

- You can use a text viewer to view the .log file in the syslog directory. System events are logged to this file. When configured for logging, both alarms and events enter the RTDB and are then stored in the Citadel historical database.

- Alarms and events logged to Citadel can be viewed with the Historical Data Viewer in the Measurement & Automation Explorer (MAX) environment. For more information about Historical Data Viewer, refer to its online help in MAX.

## Viewing Alarms and Events with the Alarm & Event Display Control

The alarm & event display is the easiest way to monitor and acknowledge alarms and events. The alarm & event display shows alarms and events from every computer and process you configure it to display. You can place the alarm & event display control on your front panel; it is located on the **Controls»Alarms and Events** palette. Another way to view the alarm & event display is in the bottom right pane of the Tag Monitor.

The alarm & event display automatically monitors all alarms generated by a process for which you are monitoring one or more tags. To monitor alarms from a process you are *not* monitoring a tag in, you must first select the source of the alarms, as follows.

1. Right-click the alarm & event display and select **Select Processes** from the shortcut menu. The **Select Processes** dialog box appears.

2. In the **Available Processes** listbox, navigate to the process for which you want to view alarms, select it, and click the **Add** button. The process appears in the **Selected Processes** listbox.

3. Click the **OK** button.

## Acknowledging Alarms in the Alarm & Event Display Control

Right-click an alarm and select an acknowledgement option from the shortcut menu to acknowledge alarms.

# Filtering Alarms and Events in the Alarm & Event Display Control

You can set filter criteria so you only see certain alarms and events in a display.

1. Right-click the alarm & event display while the VI is running and select **Filter Options** from the shortcut menu.

2. Select among the filter options.

   a. Place a checkmark in the **Priority** checkbox and type values in **Min** and **Max** to monitor alarms with specific priorities.

   b. Place a checkmark in the **User Name** checkbox and type a user name if you want to restrict alarm monitoring to alarms generated while that user is logged on. You can select only one user name at a time, but you can use asterisk (*) or question mark (?) wildcards to widen the scope of the alarms reported.

   c. Place a checkmark in the **Ack User Name** checkbox and type a user name if you want to restrict alarm monitoring to alarms acknowledged by that user. You can select only one user name at a time, but you can use wildcards to widen the scope of the alarms reported.

   d. Place a checkmark in the **Ack Comment** checkbox and type a comment if you want to restrict the alarms displayed to those with that acknowledgement comment.

   e. Place a checkmark in the **Object Name** checkbox and type an tag name if you want to restrict alarm monitoring to alarms involving that tag name. You can enter only one tag name at a time, but you can wildcards to widen the scope of the alarms reported. The tag name must be a completely qualified tag name, as displayed in the tag display pane above the alarm view.

   f. Place a checkmark in the **Description** checkbox and type a description if you want to restrict monitoring to alarms that meet your criteria. You can select only one description category at a time, but you can use wildcards to widen the scope of the alarms reported. The categories HiHi, Hi, Lo, or LoLo are added as a prefix to any descriptions and are ignored by description filtering.

   g. Place a checkmark in the **Area Name** checkbox and type an area name if you want to restrict monitoring to that alarm area. You can enter only one alarm area at a time.

   h. Use the fields in the **Old Alarms** section if you want to display alarms after they have been acknowledged.

     i.    Select a **Show** option to display alarms only, events only, or both alarms and events.

     j.    Place a checkmark in the **Audible Alarms** checkbox if you want to enable a sound alert when an alarm takes place. The sound depends on your system setting for error sounds.

3.    Click the **OK** button. The alarm & event display control displays only the alarms that meet all the filter criteria in the alarm view.

## Using an Alarm Summary Display

You can report alarms to an HMI by using the alarm summary display listbox located on the **Controls»Alarms and Events** palette. An *alarm summary* is a collection of all the alarms that currently exist in the system. In addition, if a tag previously in alarm returns to normal but is unacknowledged, a notification is posted in the alarm summary. You can invoke the HMI Wizard to create the block diagram for an alarm summary, or you can build your own block diagram.

If you are building a block diagram, use the Read Alarm Summary VI. If you want to change the default fields (time, date, tag name, alarm limit) that are visible in the alarm summary display listbox, you can use the alarm summary format control located on the **Controls»Alarms and Events** palette and change the checkbox selections. You also can change the default colors of alarms, acknowledged alarms, and unacknowledged tags that have returned to normal using the color codes for alarm summary control located on the **Controls»Alarms and Events** palette.

**Note**   The Value column displays the value of the tag when the tag first enters the alarm state, not the live value of the tag. The Value column does not update, even if the tag value subsequently changes. To monitor the changing value of a tag along with its alarm status, use the Read Tag VI and monitor the **in alarm** output to flag an alarm condition.

The alarms displayed in the alarm summary display listbox can be filtered using the HMI Wizard by group or tag names, priority, and acknowledgment status.

## Using an Event History Display

You can report recent events to an HMI by using the event history display listbox located on the **Controls»Alarms and Events** palette. An *event history* is a collection of all the alarms and events pertaining to tag values that have occurred in the LabVIEW DSC module since the Tag Engine started. You can invoke the HMI Wizard to create the block diagram code

for an event history. You also can build your own block diagram and use the Read Event History VI located on the **Functions»Alarms & Events** palette.

If you want to change the default fields (time, date, tag name, alarm limit) that are visible in the event history display listbox, you can use the event history format control located on the **Controls»Alarms and Events** palette and change the checkbox selections. You also can change the default colors of alarms, events, and normal and acknowledged alarms using the color codes for event history control located on the **Controls»Alarms and Events** palette.

The alarms displayed in the event history display listbox can be filtered using the HMI Wizard by group or tag names, priority, and acknowledgment status.

## Acknowledging Alarms in the Alarm Summary and Event History Displays

To acknowledge alarms and events displayed by the alarm summary and event history controls, create an **ACK** button by selecting a button on the **Controls»Boolean** palette and wiring it on the block diagram to the Acknowledge Alarms VI located on the **Functions»Alarms & Events** palette.

When you acknowledge these alarms, the acknowledgment status in the display changes from UNACK to ACK, and the color of the text changes from red to yellow. These are the default colors, and you can change them on the block diagram or using the HMI Wizard.

There are two modes for handling tags that were previously in alarm but have returned to normal: **Auto Acknowledge** and **User Must Acknowledge**. These modes are configured in the Tag Configuration Editor for each tag. If a tag is configured for **Auto Acknowledge**, the acknowledgment status automatically changes from UNACK to ACK when the tag returns to normal. If the tag is configured for **User Must Acknowledge**, the status remains at UNACK until the operator clicks the **ACK** button to acknowledge the alarm.

You can select the tags for which you want to acknowledge alarms. It is a good idea for this tag list to be identical to the list of tags you display alarms for in the alarm summary or event history display. For example, if you select group <ALL> in the Acknowledge Alarm VI, you can acknowledge alarms for all tags that were configured for alarms as they occur. In the tag selection, you also can select a combination of tag names and groups.

## Reporting and Viewing Alarm Status in the Alarm Summary and Event History Display

You can report the status of alarms currently in the system using the output of either Read Alarm Summary VI or Read Event History VI, or by using the Get Alarm Summary Status VI. This status includes the number of active alarms and unacknowledged alarms in the system. You can use the alarm summary status control located on the **Controls»Alarms and Events** palette to display this information on the front panel.

# Viewing System Errors and Events

*System errors* are conditions on a system level (as opposed to a tag-level basis) that result in problematic functioning of the LabVIEW DSC module. When a system error occurs, LabVIEW prompts you with a dialog box. You can turn this dialog box on or off.

*System events* are changes in the system that cause a change in behavior that is not problematic. These include events reported by utilities such as the Tag Configuration Editor.

Detailed system error and event messages are logged to a system log file. The messages are written to an ASCII file with a `.log` extension in the `SYSLOG` directory. The LabVIEW DSC module automatically creates this directory, if it does not exist already. The system log file names take the format, `YYYYMMDDHHMM.log` where `YYYY` = year, `MM` = month, `DD` = day, `HH` = hour, and `MM` = minute.

# Archiving and Extracting Alarm and Event Information

To archive alarms and events that have been logged, use the Archive Alarms and Events VI on the **Alarms** palette. Refer to the *Logging and Printing for Alarms and Events* section in this chapter for more information about logging alarms and events.

To extract alarm and event information from a Citadel database, you can use the HIST Alarm & Event Query VIs on the **Alarms & Events** function palette or the HIST Alarm & Event Query controls on the **Alarms and Events** controls palette.

# Error Handling

Errors that occur in the LabVIEW DSC module VIs can be handled in one of two ways: by the Tag Engine or by each VI. The Tag Engine handles errors for the Tags VIs and the Alarms and Events VIs. The other VIs include standard LabVIEW **error in** and **error out** parameters for error handling.

## Errors Reported by the Tag Engine

The LabVIEW DSC module reports error handling information for Tags VIs, Tag Attributes VIs, and Alarms and Events VIs to the system. If you try to access a tag that does not exist in the Tags VIs or the Alarms and Events VIs, the Tag Engine reports an error. The error shows up in the Engine Manager display. Refer to the *Viewing Tag Engine Status* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*, for more information about the Engine Manager.

## Errors Not Reported by the Tag Engine

If a LabVIEW DSC module VI does not report to the Tag Engine, it uses a standard LabVIEW control and indicator (**error in** and **error out**) to notify you that an error has occurred. Refer to the LabVIEW documentation for more information about error handling.

# 7

# Historical Data Logging and Extraction

The real-time database (RTDB) is contained in memory; there is no file created to hold the data. When the Tag Engine is stopped, the RTDB retains the last data received, but does not update any values until the Tag Engine begins to run again. Because data logged to the Citadel historical database are taken from the RTDB, no data can be logged to Citadel while the Tag Engine is stopped.

## Citadel Historical Database

The LabVIEW Datalogging and Supervisory Control (DSC) module uses the National Instruments Citadel historical database. The LabVIEW DSC module also includes the Citadel ODBC driver that has special commands to perform data transforms, so you can retrieve, manipulate, and analyze historical data automatically from outside the LabVIEW environment. Refer to Appendix A, *Using SQL to Access Historical Data in Citadel*, for more information.

Under Windows 2000/NT/XP, Citadel runs on your computer as a service, accessible through the Service Manager.

The LabVIEW DSC module installs a National Instruments services manager, denoted by an icon located in the system tray of the Windows taskbar, near the computer clock. The right-most column of circles represents the Citadel Service. A green light indicates the service is running. A red light indicates the service is stopped. To start and stop the Citadel Service, right-click the service manager icon.

⚠ **Caution** Do *not* stop these services while the LabVIEW DSC module, the Tag Engine, or Lookout is running.

Data you configure to be logged to Citadel resides in a set of files in the target directory you set for logging. This data can include values from the application as well as alarms and events. You control which data is logged to what location through tag configuration and alarm and event

configuration. You can log data to your local computer or to a remote computer on your network, but the directory to which you want to log must be writable from the computer running the Tag Engine.

You access Citadel data through the Historical Trend Viewer, Historical Data VIs, SQL queries, or any other ODBC-compliant application such as Microsoft Query, Microsoft Access, or even Microsoft Excel.

# Logging Historical Data

Complete the following steps to log historical data.

1. Make sure you have configured your tag(s) for logging as described in the *Configuring a Tag to Log Data or Events* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*.

2. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

3. Select **Configure»Historical** to display the **Historical Logging Configuration** dialog box.

4. Select among the historical logging options. To open the context help window and view descriptions of these options, press <Ctrl-H> and move the cursor over any field.

**Note**   If you log data to directories created in a secure file system, such as NTFS, you must grant the System account Change or Full Control permissions to the directory. If you do not grant the System account appropriate access to the database directory, Citadel will be unable to create and modify the database files it uses to store historical data and alarms.

5. Click the **OK** button.

6. Select **File»Save** to save the changes.

7. Open the Engine Manager (with the Tag Engine running, select **Tools»Datalogging & Supervisory Control»Launch Engine**).

8. Place checkmarks in the **Log** checkboxes to turn on historical logging.

   You also can use the Enable Historical Data Logging VI located on the **Functions»System** palette to turn historical data logging on or off dynamically for all the tags in the system, while the Tag Engine is running.

# Logging Data in Sets

To log and retrieve data in sets, you configure the Data Set Logger server to track your data sets, then use either the Historical Data Viewer in Measurement & Automation Explorer (MAX), or the Data Set Logger VIs, to retrieve your data set values. With the Data Set Logger, you can accomplish *batch logging*.

A group of tag values that are logged together during some finite time period is a *data set*. A data set might cover a batch of some sort and contain all the values generated during a single execution of a batch process. The *ID tag* for each data set denotes a particular data set and the time during which the data set run took place. An ID tag might be a batch number.

The Data Set Logger is a VI-based server that allows multiple data sets to execute simultaneously. It starts when the Tag Engine starts and stops when the Tag Engine stops, but executes only when a .scf file uses it.

## Creating a Data Set for Logging

To create a data set for logging, follow these steps.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**) and make sure the active .scf file contains the tags you want to log in your data set.

2. Save your .scf file if you have added or removed tags since the last time you saved.

3. Select **Servers»Data Set Logger Configuration**. The following dialog box appears. If the dialog box opens with existing data set values, you can edit that data set or select **File»New** to create a new data set.
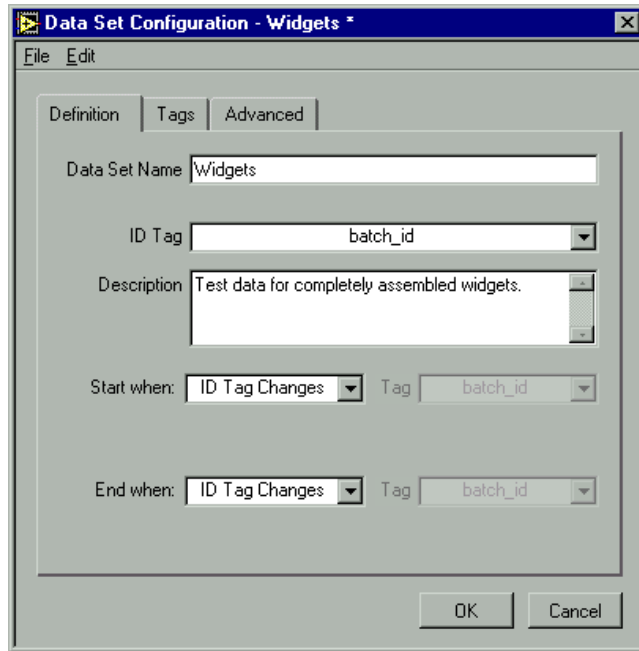
4. Enter values on the **Definition** tab.

   - **Data Set Name** is a string that provides information about the data being grouped together. You can use any number of different data set configurations, but each one must have a unique name.

   - **ID Tag** is a string or analog tag from your active .scf file. When you start logging a data set, the value of the **ID Tag** at that time becomes the identifier of that data set run. For example, an **ID Tag** might be the serial number of a unit under test, and the data set for that serial number might consist of the traces logged during testing of that unit. Because the same unit might be tested more than once, that **ID Tag** might be used more than once. The Data Set Logger handles this case by creating a unique internal ID for every run. However, you might want to assign a unique **ID Tag** for each run. For example, you could combine a serial number with a timestamp.

**Note**    Analog ID tags are treated as double precision numbers when stored in the database.

   - **Description** is text information about the configuration.

- **Start when** and **End when** fields and their associated **Tag** fields specify the type of start/end condition, and a tag to monitor for fulfillment of that condition. When the start condition is met, a new run starts, provided that the previous run has ended. When the end condition is met, the run ends. The following options are available. The numbers can be used to programmatically change the start and stop conditions.

**Table 7-1.** Data Set Run Start/End Conditions

| Value | Start/End Condition |
|-------|---------------------|
| 0 | **ID Tag Changes**—when the value of the **ID Tag** changes, a new run starts. |
| 1 | **Discrete Tag ON**—when the specified discrete tag's value changes from FALSE to TRUE, a new run starts. |
| 2 | **Discrete Tag OFF**—when the specified discrete tag's value changes from TRUE to FALSE, a new run starts. |
| 3 | **Analog Tag > Limit**—when the specified analog tag's value exceeds the user-provided limit, a new run starts. |
| 4 | **Analog Tag = Limit**—when the specified analog tag's value equals the user-provided limit, a new run starts. Be careful when using the **Analog Tag = Limit** setting, because comparisons are done with floating point numbers. For example, 6.9 does not equal 6.90001 with this option. |
| 5 | **Analog Tag < Limit**—when the specified analog tag's value is less than the user-provided limit, a new run starts. This option uses analog tags only. |
| 6 | **String Tag = Value**—when the specified text string tag's value equals the user-provided string, a run starts. Use only text strings for this option. |
| 7 | **Time of Day**—when the system's clock reads the specified time of day (0:00:00 to 23:59:59), a new run starts. No tag is used with this option. |

5.  Click the **Tags** tab, and click **Add** to select the tags you want included in your data set. All tags/traces to which a data set refers must be logged in the same database. To remove tags, select the tags you want to remove, and click on the **Remove** button.

**Note**    You should use a data set configuration only with the .scf file you used to create it. Additionally, if you change the name of a tag in your .scf file, and that tag is used in a data set configuration, you must edit the data set configuration separately. Changes to a .scf file will not show up in the data set configuration tool until the .scf file is saved.

6.  (Optional) Click the **Advanced** tab, and click **Add** to enter an item and a description of the equipment used during the data set run. This information is stored as text strings, with each new run.

7.  Click **OK**.

8.  Create at least one tag connection from your .scf file to a Data Set Logger server item. This ensures that the Data Set Logger server will run, because it will be launched by the Tag Engine.

    a.  To convert a server item to a tag using the Tag Configuration Wizard, click the Configuration Wizard button in the Tag Configuration Editor. Each different data set configuration, with items, appears as a device under the Data Set Logger server, as shown in the following illustration.
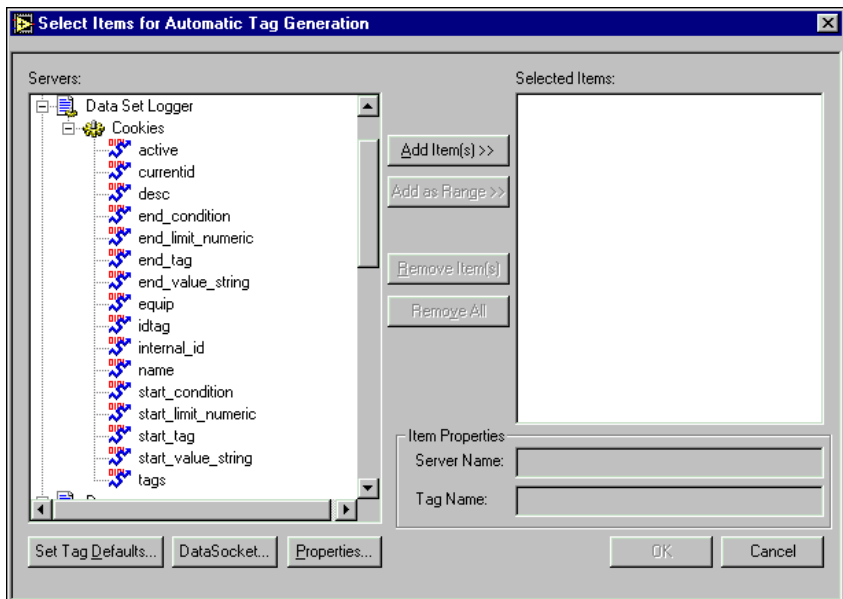
b.  Highlight the items you want to create tags from under your data set on the left, and click **Add Items**. Create at least one tag from the server items. You may find the **active**, **currentid**, and **internal_id** server items are most useful.

The Data Set Logger items are described in the following table.

**Table 7-2.**  Data Set Logger Server Items

| Item Name | Type | Description |
|---|---|---|
| active | read-only discrete | Value is TRUE after the start condition for a data set run has been met, until the end condition has been met. |
| currentid | read-only string | Reports the ID Tag value for the current data set run, while the data set run is active. |
| desc | read-write string | Contains the description of the data set. If written to, the description is updated at the start of the next data set run. |
| end_condition | read-write analog | Specifies the type of end condition to use for the next data set run. The allowed values are listed in Table 7-1. Depending on the end condition, certain other items may need to be updated as well. |
| end_limit_numeric | read-write analog | When end_condition uses numeric comparisons, this value is compared to the value of end_tag to determine if the end condition has been met. |
| end_tag | read-write string | Tag used to test for the end condition. The data type must be compatible with the type of condition. |
| end_value_string | read-write string | If the end_condition is configured as String Tag=Value, this value is compared to the value of end_tag to determine if the end condition has been met. The comparison is case sensitive. Use a text string for this value. |
| equip | read-write string | List of equipment for the current data run. You can write either a spreadsheet-style string (columns separated by tab), rows by EOL, or a flattened 2-dimensional array of LabVIEW strings to this item. Changes take effect when the next run starts. |

**Table 7-2.** Data Set Logger Server Items (Continued)

| Item Name | Type | Description |
|---|---|---|
| idtag | read-write string | If written to, the data set configuration is modified to use this tag as the ID Tag when the next data set run starts. |
| internal_id | read-only binary string | Contains an internally generated, 8-byte binary identifier that is unique for each data set, used to identify each data set run. You can use the Data Set Internal ID String to ID Cluster VI to convert this binary information into a displayable data structure in LabVIEW. |
| name | read-only string | Name of the data set configuration. |
| start_condition | read-write analog | Specifies the type of start condition to use at the beginning of the next data set run. The allowed values are listed in Table 7-1. Depending on the start condition, certain other items may need to be updated as well. |
| start_limit_numeric | read-write analog | When start_condition uses numeric comparisons, this value is compared to the value of start_tag to determine if the start condition has been met. |
| start_tag | read-write string | Name of the tag used to test for the start condition. The data type must be compatible with the type of condition. |
| start_value_string | read-write string | If the start_condition is configured as String Tag =Value, this value is compared to the value of start_tag to determine if the start condition has been met. The comparison is case sensitive. Use a text string for this value. |
| tags | read-write string | Provides the list of tags used in the data set, delimited by EOL (end-of-line) characters. If written to, the set of tags included in a data set will be used when the next data set run starts. You can write either an EOL-delimited text string, or a flattened LabVIEW array of strings to this tag. |

When you connect to writable server items, you can programmatically configure those items from your VI. All writable items can be changed at any time, but values changed during a run do not take effect until the run ends.

c. Add server items from each data set you have defined. Click **OK** when you are finished.

9. Restart the Tag Engine if it is running, so that your changes take effect.

## Editing Data Sets for Logging

To edit an existing data set, follow these steps.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**) and make sure the active `.scf` file contains the tags you want to include in your data set.

2. Select **Servers»Data Set Logger Configuration**. If the dialog box does not open to the data set you want to edit, select **File»Open** to open the data set you want to edit.

3. Make any changes that you want to make to the configuration of the data set. Refer to the *Creating a Data Set for Logging* section of this document for more information about the configuration options.

4. Restart the Tag Engine if it is running, so that your changes take effect.

## Considerations for the Data Set Logger

Nested data sets are not allowed. That is, you cannot include a data set within another data set. Also, if a data set starts but does not properly meet its end condition, it is an open-ended run and will not appear as a complete data set run when accessing completed data sets.

## Retrieving Logged Data Sets

You can use either the Historical Data Viewer in MAX, or the Data Set Logger VIs, to retrieve data that has been logged in sets using the Data Set Logger server. Refer to the Historical Data Viewer documentation in MAX for more information about it. Refer to the LabVIEW DSC module online help for more information about the Data Set Logger VIs.

# Archiving Historical Data

You can archive historical data in the following ways.

- Use the archiving feature of Historical Data Viewer in Measurement & Automation Explorer (MAX).

- Use the Archive Database VI on the Historical Data palette. This VI also allows you to merge two existing databases.

- Archive the database manually, by copying or moving the files. When you decide to archive historical data manually, copy the .scf file along with the historical files to the new location. Although you can retrieve historical data without the .scf file, you do not have the tag configuration information, such as engineering range and unit, unless you archive the .scf file as well. You *must* stop the Tag Engine and Citadel Service before archiving these files manually.

  When you create a .scf file, the default location for the data generated by the tags configured in that file is a directory called data located in the directory in which you saved the .scf file. Preferably, maintain the relative path between the .scf file and the historical files in the new archive location. For example, if the .scf file is in c:\archive, keep the historical database in c:\archive\data.

  You need to archive files with the following extensions: .ale, .adx, .dat, .mdx, .bak, .tbd, .tdx, .thd. These files are not independently accessible. Put these files into the folder you have selected as the logging directory. However, if you copy those files into a folder with an already existing database, the files names will collide. The only way to safely merge two databases is with the Archive Database VI.

# Converting Older Citadel Database Files

If you logged to Citadel database files from BridgeVIEW or Lookout versions earlier than 4.0, you must decide whether to update the older files to the current Citadel format. You can convert old database files to maintain data continuity, or you can start a new data directory and keep your old data segregated from new data.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**). When you first open a tag configuration file generated in BridgeVIEW, or set the logging directory to a directory containing an old database file, you are prompted to convert the file.

2.  Select **Configure»Historical** and set the **Data Directory** to the path
    to the old file, or select **File»Open** to open the `.scf` file associated
    with the old file. The following dialog box appears.
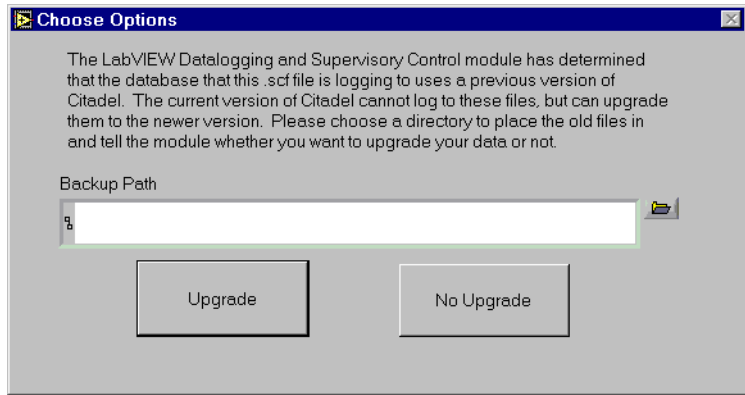


3.  Choose whether to convert the files.

    •   Click **Upgrade** to convert your old data files to the new format.
        The time required for LabVIEW to convert your files depends on
        the amount of data. The updated data will show that all tag values
        were produced on the computer on which you are performing the
        conversion. New data logged will show the proper source for the
        data.

    •   Click **No Upgrade** if you do not want to convert the files.
        LabVIEW will create a new, empty database in your current data
        directory for logging data in the new format.

Whether or not you choose to convert your files, LabVIEW moves your old
files to the path you enter in the **Backup Path** field. If you leave this field
blank, LabVIEW moves the files to a subdirectory of your data directory
called `archive`.

# Extracting and Viewing Historical Data

There are five methods for viewing historical data that has been logged to
Citadel.

•   Use the Historical Data VIs and Historical Trend display. The
    Historical Trend control on the **Trends** control palette is similar to the
    standard LabVIEW XY graph, so it utilizes a traditional LabVIEW
    programming approach. It is intended for use on a front panel when
    you want to display graphical data that is being acquired with

Historical Data VIs. Because you can only display nine traces, it can be a more memory-intensive method for trending data, if that is all you want to do.

The traces that the Historical Trend control displays are specified on the block diagram. You cannot browse to other traces, zoom in, jump to breaks, or jump to minimums and maximums. Refer to the following section, *Accessing Historical Data Using Historical Data VIs*, for more information.

- Use the classic Historical Trend Viewer (HTV). The HTV is a utility that launches as a separate window in the LabVIEW DSC module environment. It is one of the easiest ways to display a historical trend, and is configurable from the block diagram. You can launch it from the **Tools** menu or use the Call HTV.vi to load the HTV with a given set of traces and ranges.

  The HTV display is limited to eight traces at once, but you can browse to different traces within the current `.scf`. The HTV hypercursor allows you to locate trend breaks, but not minimums or maximums. Refer to the *Accessing Historical Data Using the Historical Trend Viewer* section later in this chapter for more information.

- Use the Historical Data Viewer. The Historical Data Viewer exists in the Measurement & Automation Explorer (MAX) environment and requires no programming. The Historical Data Viewer allows multiple views (collections of traces and settings) that can be saved for future viewing. These views can be called up programatically from a VI, but must be configured in MAX.

  With the Historical Data Viewer, you can view any number of traces and browse to any traces within a single database. You can zoom out to any width, locate breaks, and jump to minimums and maximums of a trend. The Historical Data Viewer is documented in MAX.

- Use the NI HyperTrend control on the **Trends** control palette. The new NI HyperTrend control is a more flexible, high-performance trending tool that may require additional programming. You can drag it from the **Trends** control palette, place it on a front panel, and customize it in the user interface or in code. HyperTrend control configuration is saved with the VI that contains it.

  With the NI HyperTrend control, you can view any number of traces and browse to traces from multiple databases. You can zoom out to any width, locate breaks, and jump to minimums and maximums of a trend. Access HyperTrend control documentation by right-clicking the HyperTrend chart on a front panel and choosing **Help**.

• Use an ODBC-compliant program to query the Citadel database. Refer to Appendix A, *Using SQL to Access Historical Data in Citadel*, for more information.

## Accessing Historical Data Using Historical Data VIs

There are several VIs you can use in an HMI to manipulate data logged in Citadel files. These VIs access disk files and do not require the Tag Engine to be running. You can use these VIs to browse files, display data in a Historical Trend, export the data to a spreadsheet, or perform various database administrative tasks. There are several examples in `labview\examples\lvdsc` to illustrate the use of these VIs.

The following illustration shows a VI for viewing historical data files anywhere in the system.
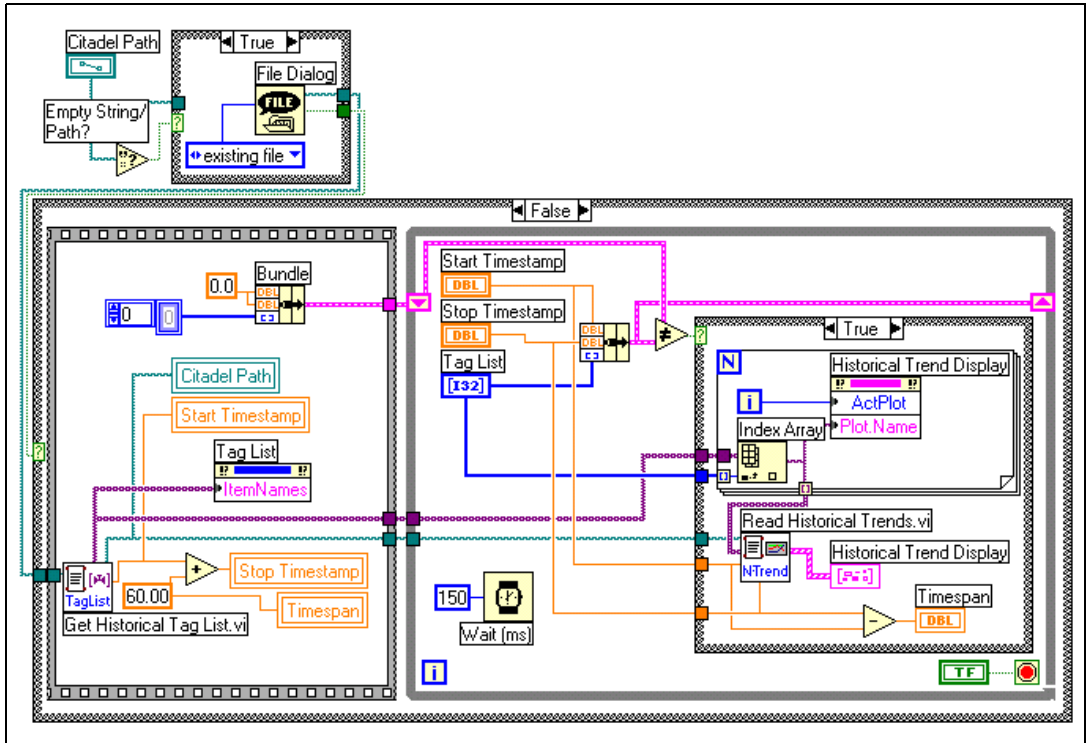


The example uses the **history tag list** returned by the Get Historical Tag List VI to set up a list of names in the front panel array of tag controls. It uses the **first timestamp** output to initialize the **Start Timestamp** control on the front panel. By default, the example displays the first 60 seconds

worth of data on the historical data display. Historical data are displayed using the Historical Trend XY graph located on the **Controls»Trends** palette.

The Read Historical Trends VI returns historical data from **Start Timestamp** to **Stop Timestamp** for the tags that are selected in the Tag List listbox and returns data in a form that can be wired directly to an XY graph.

The shift register in the While Loop is used to detect user input changes on the operator interface by remembering the previous **Start Timestamp**, **Stop Timestamp**, and the selected **tag list**. If any of these controls are changed, historical data are retrieved for the new settings and the historical trend XY graph is updated.

## Using URLs with Historical Data VIs

You can use URLs as well as tag names as the **Tag Name** input to the Historical Data VIs. A URL is the way Logos networking protocol locates data. If you are using a tag control in your application and want to support the ability to type in a URL from the front panel directly into a tag control, right-click on the tag control and enable **Allow Undefined Tags**.

The general form for the URLs for LabVIEW DSC module tags is

`\\computername\ProcessName\tagname`

where `computername` is the name of the computer running the Tag Engine containing the `tagname` on your network and `tagname` is the name of the data point you want to connect to.

When you are connecting to a data source in Lookout or in some other National Instruments data source using the Logos networking protocol, you use the URL address element `ProcessName`, where `ProcessName` is the name of the process containing the data point you want to connect to.

If you use a URL input for **Tag Name** in addition to a data path for **Citadel Path In**, the LabVIEW DSC module checks the directory specified by **Citadel Path In** for the tag part of the URL. If the LabVIEW DSC module does not find that tag, it reports that it cannot access the tag.

If you use a URL input for **Tag Name** while the **Citadel Path In** input is blank, the LabVIEW DSC module uses the URL to locate the tagged data.

# Accessing Historical Data Using the Historical Trend Viewer

The Historical Trend Viewer (HTV) is a stand-alone utility that enables you to look at historical data in your system. The HTV limits you to viewing no more than eight tags at a time, and you can view data from only one Citadel database at a time. If you want to look at more tags or multiple databases in a single historical trend, you should use another method such as the Historical Data Viewer or NI HyperTrend.

To open the HTV, select **Tools»Datalogging & Supervisory Control» View Historical Data**. The HTV is shown in the following illustration.



| 1 | Stop time | 5 | Autoscale Y button | 8 | Cursor Movement tool | 11 | Panning buttons |
|---|-----------|---|--------------------|---|----------------------|----|-----------------|
| 2 | Start time | 6 | Scrolling tool | 9 | Trend display | 12 | Time axis |
| 3 | Y-axis displays | 7 | Zoom button | 10 | Plot legend | 13 | Cursor displays |
| 4 | Full Scale button | | | | | | |
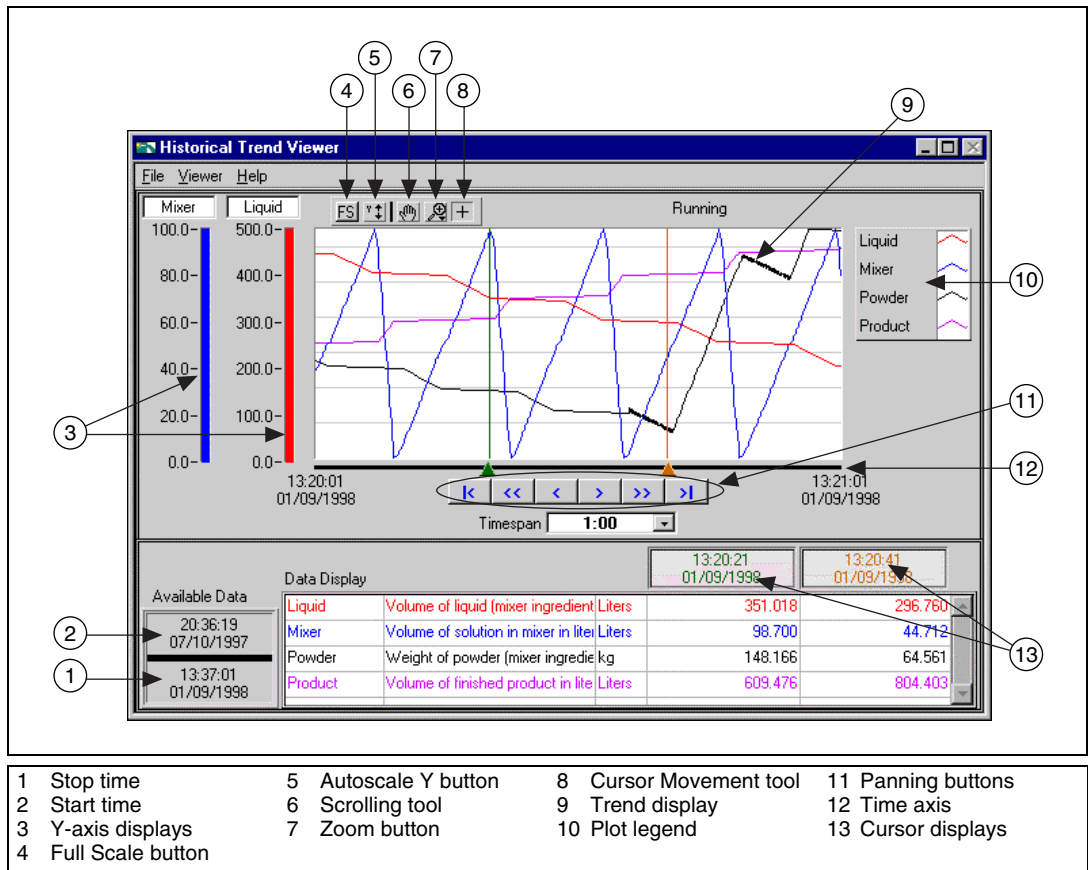
**Figure 7-1.**  Historical Trend Viewer Environment

## Selecting Tags to Display in the HTV

1.  Open the HTV. The **Select Tags for HTV** dialog box appears. If the HTV is already open, select **File»Select Tags** to display the dialog box.

2.  Enter a **Logging Computer**. The default is localhost, but you can browse to and select any registered computer on your network that is running Citadel.

3.  Select either a .scf file or a directory of Citadel files. The default is to select a .scf file. The .scf file you select must point to a valid directory of Citadel files. If the Tag Engine is running, the .scf file being used by the Tag Engine is selected by default.

4.  Select the tags you want to display in the **Available Tags** listbox and click the **Add** button to display them in the **Tags to Display** listbox on the right. The HTV displays the tags in the order that they are listed in the **Tags to Display** list.

    View configuration information about a tag by selecting it in the **Available Tags** listbox and clicking the **Tag Information** button.

5.  Click the **OK** button.

## Changing the HTV Time Axis with Panning Buttons

You can change the time axis for a trend within the HTV manually, or by using panning buttons. The panning buttons allow you to move backward and forward through the historical data in the trend. The buttons do not affect the timespan of the trend. For example, if the trend displays data from 9:45 to 9:55 on the same day, the timespan is ten minutes. Table 7-3 describes the panning button functions.

**Table 7-3.** Panning Button Functions

| Button | Name | Description |
|--------|------|-------------|
| \|< | Retrieve oldest data | Displays the first available page of data. |
| << | Back to closest point | Centers the display around the closest point to the left of the timespan. If there is no data in the previous time span, skips to the previous end of data. |
| < | Back one-half page | Moves the display back by half of the current timespan. |
| > | Forward one-half page | Moves the display forward by half of the current timespan. |

**Table 7-3.** Panning Button Functions (Continued)

| Button | Name | Description |
|--------|------|-------------|
| >> | Forward to closest point | Centers the display around the closest point to the right of the timespan. If there is no data in the next time span, it skips to the next start of data. |
| >| | Most recent data | Displays the most recent available page of data. |

## Changing the HTV Time Axis Manually

You also can change the time access for a trend within the HTV manually. To do so, select the text at either end of the time axis and change the data. You must enter the date in the correct format. If you make an error, the input is ignored.

You can select and enter the time and date on the time x-axis of the historical trend on the HTV directly. However, the HTV responds immediately to any changes you make. If you want to make manual edits to both the start and stop time on the time axis, you can select the **Viewer»Time & Date** option. When you select this option, the following dialog box appears and you can enter the start and stop time of the data displayed in the trend.



## Changing the HTV Timespan of Data Displayed

The **Timespan** pull-down menu displays the amount of relative time between the start and end points of the time axis. To change the amount of time between these points, you either can manually reenter data in the start or end point on the time axis, or use the **Timespan** pull-down menu.

By default, **Timespan** contains the values 1:00, 5:00, 10:00, and 30:00. Select **Enter New** in **Timespan** if you want to enter a different amount of data to display.

# Viewing an HTV Tag Value at a Specific Point in Time

The Data Display table on the HTV shows the tags displayed in the trend, the tag description, and, for analog tags, the engineering units associated with the tag. The two rightmost columns show the values of the tags at the two cursor locations in the trend. For discrete tags, the values in these columns are either `On` or `Off`. To move the cursors, drag the triangles at the bottom of the trend display.

# Changing the HTV Y-Axis

The HTV displays two Y axes at any time. Each y-axis displays the color of the tag associated with it. All discrete tags show their ranges as going from `On` to `Off`. Click the y-axis to make it rotate through the tags displayed in the trend.

To change the range in the y-axis for analog and bit array tags, select the text at the top or bottom of the scale and type in the desired value. When you enter the value, that trend scale changes and the trend display updates. Discrete tags are displayed without y-axis scales, and ranges are shown as `On` or `Off`.

# Changing the HTV Plot Colors and Style

Click the **Trend Legend**. The shortcut menu contains several options with which you can change the plot colors and styles used in the trend.

# Zooming In on an HTV

The **HTV Trend** palette contains a **Zoom** button, shown at left, that allows you to zoom in on points of interest.

Click the Zoom button and select from the following options, clockwise from the top left, to zoom in and out of the trend:

- **Zoom to Rectangle**—Click a point on the display you want to be the corner of the zoom area and drag the tool until the rectangle covers the zoom area.
- **X-zoom**—Zooms in on an area of the graph along the x-axis.
- **Y-zoom**—Zooms in on an area of the graph along the y-axis.
- **Zoom Out about Point**—Click a point you want to zoom out from.

- **Zoom In about Point**—Click a point you want to zoom in on. Press the <Shift> key to change between Zoom In about Point and Zoom Out about Point.

- **Undo Zoom**—After you zoom in or out, use this option to return to the previous view.

## Exporting HTV Data to a Spreadsheet

From the HTV, select **File»Export**. The HTV exports the information currently displayed in the trend to a tab-delimited file. A dialog box prompts you for the name and location of the file to create.

The HTV resamples data in periodic intervals so that all tags have the same number of data points. The frequency defaults to a value according to the frequency of data in the historical files. If you want to override this value, enter the frequency you want in the dialog box.

## Setting HTV Tag, Time, and Color Preferences

Set the preference for the HTV to remember settings for display time and color on exit by selecting **Viewer»Preferences**. When you exit the HTV, the state of the viewer is recorded.
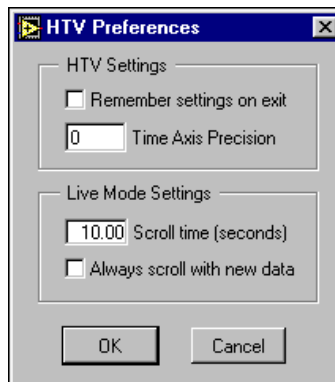


Place a checkmark in the **Remember settings on exit** checkbox if you want to update your settings each time you exit the HTV.

## Viewing Newly Logged HTV Data Automatically

You can use Live mode to watch incoming data after it has been logged. When historical logging is enabled, the **Live** button appears to the right of the panning buttons. When you click the **Live** button, the trend

automatically updates periodically. Select **Viewer»Preferences** in the
**HTV Preferences** dialog box to set how often the trend displays the new
data. The default is 10 seconds. If **Always scroll with new data** contains a
checkmark, the display updates when new data are logged.

While Live mode is turned on, the values for each tag are extrapolated to
the last time the trend was updated. These extrapolated values are marked
with an asterisk in the Data Display. When a cursor or slider is placed
before the extrapolation begins for a tag, the asterisk is not present. Turning
off Live mode also turns off extrapolation.

### Calling the HTV from an Application

In many HMI applications you might elect not to give the operator access
to the standard menu bar. You can use the Call HTV VI located on the
**Functions»Historical Data** palette to call the HTV dynamically from an
HMI application.

# Printing Historical Data

You can print logged, historical data in the following ways:

- Print historical data trends from the HTV. Open the HTV by selecting
  **Tools»Datalogging & Supervisory Control»View Historical Data**.
  Refer to the online help for information about selecting tags, adjusting
  the timespan, and printing data.

- Print historical data trends from the Historical Data Viewer in MAX,
  which allows you to export data to a spreadsheet or to HTML format.

- Print historical data from a spreadsheet. To export historical data, use
  the Historical Trends to Spreadsheet VI or Historical Trends to
  Spreadsheet File VI, or export data from the HTV. Then print the data
  from the resulting spreadsheet file. Refer to the *Logging Historical
  Data* and *Extracting and Viewing Historical Data* sections for more
  information about logging and exporting historical data.

- Print historical data from a text file that you create by exporting to a
  spreadsheet file.

- Use an ODBC-compatible application to query the Citadel historical
  database and print the results. Refer to Appendix A, *Using SQL to
  Access Historical Data in Citadel*, for more information about using
  ODBC applications with the Citadel database.

**8**

# Security

To implement security in an application, set up user and group accounts. Then configure and restrict access to the LabVIEW environment and your specific HMI.

A system with permission-based security is a system in which users are allowed various degrees of access to tools or data depending on the permission attached to their account name in the access property of the tool or data involved.

# Creating and Editing User and Group Accounts

You use the User Account Manager to create and edit the properties of groups, create or edit the properties of user accounts, assign users to one or more groups, and otherwise manage security accounts for LabVIEW and Lookout applications. Only an administrator or someone whose account is a member of the Administrator group can create, revise, or delete system user accounts.

✎ **Note** For user accounts to work consistently across your network, you must use the same lookout.sec file for all installed copies of Lookout or the LabVIEW Datalogging and Supervisory Control (DSC) module. Refer to the *Duplicating Security Files for Networked Computers* section of Chapter 9, *Networking and Deploying Applications*, for more information.

## Creating User Accounts

1. Open the User Account Manager (**Tools»Datalogging & Supervisory Control»Security»Edit User Accounts**.)

2. Select **User»New User**.

3. Enter the domain name of the new user in the **Username** field.

4. Enter the **Full Name** of the user.

5. Enter job titles or other relevant information in the **Description** field.

6. Enter the user password in the **Password** field.

7.  Enter the password a second time in the **Confirm Password** field to make sure there was no typing error in the first entry.

8.  Set the **Security Level** for the new user. Security levels range from 0 to 10, with 10 being the highest possible security authorization. Assign level 10 access only to those people responsible for system security.

9.  **Minutes idle until logoff** sets how long LabVIEW runs with no operator interaction before logging the current user off automatically. Setting this value to 0 (the default) means there is no timeout in effect.

10. You can set an expiration time for passwords. Users cannot reset their own password; a member of the Administrator group must set the password for them. The default is for passwords never to expire.

11. Place a checkmark in the **Account Disabled** checkbox if you want to disable a user account without removing the user from the system.

12. Click the **Groups** button to add this user to various local security groups. The **Group Memberships** dialog box appears.

    The default groups are Administrators, Guests, Operators, and System Operators. Any groups you have created are also shown.

13. To enter a user in a group, highlight the group in the **Not Member of** field and click the **Add** button. To remove a user from membership in a group, highlight a group in the **Member of** field and click the **Remove** button.

**Note**  When you add an individual user whose individual account has a security level different than that of the group, that user has the higher of the security levels.

14. Click the **OK** button.

# Creating Groups

1.  Open the User Account Manager (**Tools»Datalogging & Supervisory Control»Security»Edit User Accounts**.)

2.  Select **User»New Local Group**.

3.  Assign a name to the group in the **Group Name** field.

4.  Enter a description of the group in the **Description** field.

5.  Assign the security level for members of this group in the **Security Level** field.

**Note**  When you add an individual user whose individual account has a security level different than that of the group, that user has the higher of the security levels.

6. To add **Members**, click the **Add** button. The **Add Users and Groups** dialog box appears.

7. The **List Names From** listbox selects the domain to list user names from. At this time, you are restricted to your local domain.

8. Highlight the names you want to add in the **Names** field, and click the **Add** button to add those users to your group.

## Modifying User and Group Accounts

The dialog boxes for editing users and groups are essentially the same as those for creating users and groups. Complete the following steps to modify user and group accounts.

1. Open the User Account Manager (**Tools»Datalogging & Supervisory Control»Security»Edit User Accounts**.)

2. Either double-click the user or group you want to edit, or highlight the user or group and select **User»Properties**. The **User Properties** dialog box appears and displays information about user activity.

3. Use the **User Properties** dialog box as you would the new user dialog box. Refer to the *Creating User Accounts* section for more information about the fields in this dialog box.

4. Click the **OK** button.

## Special Pre-Defined User and Group Accounts

The National Instruments User Account Manager comes with several user accounts and groups built-in. The built-in user accounts include Administrator, Everyone, Guest, and (nobody). The built-in groups include Administrators, Guests, Operators, and System Operators. You cannot delete any of these accounts, though you can edit the properties of some of them.

The Administrator account overrides all other security settings and has access to everything in LabVIEW. This override extends to all individual accounts added to the Administrators group.

You cannot delete the Administrator account or change its security level. You can set the password and enter the name and a description of the Administrator. You can add or remove individual user accounts from the Administrator group.

The (nobody) account cannot be edited or deleted, and does not actually appear as an account in the User Account Manager. This account is what LabVIEW defaults to when no authorized user is logged on. It always has a security level of 0.

You can edit all the properties of the Guest user account and of the Guests, Operators, and System Operators groups.

## Logging In and Out

To log in, select **Tools»User Name**. Type your account name and password. If you do not know your account name, or have forgotten your password, contact your LabVIEW administrator.

To log out, select **Tools»Datalogging & Supervisory Control» Security»Logout**, or select **Tools»User Name** and click **Logout**.

## Accessing User Information

After you log into the LabVIEW DSC module, you can find out what your user privileges are, along with other user information, by selecting **Tools» Datalogging & Supervisory Control»Security»User Info**.

The default tab for the **User Information** dialog box lists the identity of the logged in user along with activity information for this user. Other tabs reveal the permissions set for a given user.

## Changing Your Password

Administrators can change passwords by editing accounts in the User Account Manager. If you are not an Administrator, follow these steps.

1.   Make sure you are logged in (**Tools»User Name**).

2.   Select **Tools» Datalogging & Supervisory Control»Security»Change Password**.

3.   Type your old password, your new password twice, then click **OK**.

# Restricting Access to the LabVIEW Environment

After you set up your user and group accounts, you can implement security in several ways. You can configure access to most LabVIEW DSC module utilities and the Tag Engine on a per-user or group basis. HMI VIs can enforce security by determining whether the current user can operate, or even see, a particular control or indicator. In general, security set up by

selecting **Tools»Datalogging & Supervisory Control»Options** or
**Tools»Datalogging & Supervisory Control» Security** applies to
everything in the LabVIEW environment, security set up in the Tag
Configuration Editor applies only to the `.scf` file, and security
implemented on a block diagram applies only to that application.

# Setting Access to Front Panel Controls and Indicators

To control access to a front panel object, right-click the object and select
**Security** from the shortcut menu to display the **Access Rights** dialog box.
Add to the list all the individual user accounts or groups of users you want
to have access to that control. You can then set the access level each user
or group is allowed, from full to no access.

# Setting Permissions for Accessing Tools

Complete the following steps to set permissions for the Tag Configuration
Editor, Tag Engine, Historical Trend Viewer, Tag Monitor, Server
Browser, startup VIs, or server tester.

1. Log in as an administrator (**Tools»User Name**).

2. Select **Tools»Datalogging & Supervisory Control»Options**.

3. Click the **Advanced** tab.

4. Click the **Tools Access** button. The **Tools Access** dialog box appears.

5. Click the tab for the tool for which you want to configure permissions
   and click the **Edit** button. The standard LabVIEW DSC module
   **Access Rights** dialog box appears with the name of the tool for which
   you are configuring permissions above the list of users and groups.

6. This dialog box displays a list of access rights for specific users and
   groups.

   • To remove a user or group, select it and click the **Remove** button.

   • To change a user or group permission, select it and select from the
     options in the **Access** listbox.

   • To add a new user or group, click the **Add** button. The **Add Users
     and Groups** dialog box appears. Highlight the user or group you
     want to add and click the **Add** button. Set the **Access rights**. Click
     **Help** in the **Access Rights** dialog box for detailed information.

7. Click **OK** when you are finished.

# Configuring Access to a Specific Tag

Complete the following steps to configure access to a specific tag.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Double-click the tag to display the **Tag Configuration** dialog box.

3.  On the **General** tab, click the **Security** button to display the **Access Rights** dialog box.

4.  Set the **Access rights**. Click **Help** in the **Access Rights** dialog box for more information.

5.  Click the **OK** button twice and save your changes.

📝  **Note**   If you do not use a specific setting for a tag, it inherits the data access settings made for the `.scf` file. For more information, refer to the *Setting Data Access* section later in this chapter.

# Setting SCF File Access

You can specify who can edit a particular `.scf` file. This permission is part of each `.scf` file and can vary from file to file.

1.  Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2.  Select **Configure»Security**.

3.  Click the **SCF File Access** tab, and click the **Edit** button. The **Tag Configuration File Access Rights** dialog box appears.

4.  The large field in the center of this dialog box lists the groups and user accounts that have permission to work with the `.scf` file shown at the top of the dialog box.

    *   To remove a user or group, select it and click the **Remove** button.

    *   To change a user or group permission, select it and select from the options in the **Access** listbox.

    *   To add a new user or group, click the **Add** button. The **Add Users and Groups** dialog box appears. Highlight the user or group you want to add and click the **Add** button. Set the **Access Rights**. Click **Help** in the **Access Rights** dialog box for more information.

5.  Click **OK** when you are finished.

# Setting Data Access

You can specify which users, groups, or computers are allowed to access a `.scf` file's tag data from DataSocket, Tag Monitor, Lookout, or other Tag Engines. You can also set up a proxy user account so that LabVIEW knows how to handle unidentified clients attempting to connect to tag data. The settings you make may depend on the programs and users that may attempt access, and under what circumstances.

The LabVIEW DSC module implements the following priority when checking access rights:

1. Does the computer have access? If not, access is denied. If so, the user access rights are verified.

2. Is the user recognized?

   a. If the same `lookout.sec` file exists on both the local and remote computers, the user is recognized. If the user is recognized, access rights are assigned based on the configuration of that user's account.

   b. If the user is not recognized, the proxy user settings are used.

The following situations are examples of how the LabVIEW DSC module handles various access attempts:

- A DataSocket connection to tag data: On a front panel, the access of the user currently logged in to LabVIEW is used. On a diagram, the "nobody" account is used, so the proxy user access rights are invoked.

- A Lookout user or LabVIEW DSC module user attempts access to your tag data from across the network, and both machines have the same `lookout.sec` file installed: In this case, your LabVIEW DSC module application applies the access rights assigned to that user's account.

- A Lookout user or LabVIEW DSC module user attempts access to your tag data from across the network, and both machines do *not* have the same `lookout.sec` file installed: In this case, the user is unrecognized by your LabVIEW DSC module application, and the proxy user access rights are applied.

- Someone using a program other than Lookout or the LabVIEW DSC module attempts to access your tag data from across the network: In this case, the user is unrecognized by your LabVIEW DSC module application, and the proxy user access rights are applied. This would also apply to a LabVIEW user without the LabVIEW DSC module installed.

- A separate Tag Engine connects through Logos networking to tags in your local Tag Engine: In this case, your LabVIEW DSC module application applies the access rights assigned to the *engine user* account as defined in the separate Tag Engine.

- The Tag Monitor is running: Tag Monitor uses the access rights of the user logged in whenever the Tag Monitor was launched. If the Tag Monitor is left running while a different user logs in to LabVIEW, the previous user's access rights remain in effect.

## Setting Network Access for Specific Users, Groups, or Computers

You can grant or deny tag data access across the network for users, groups, or computers. All tags in a given .scf file inherit these settings, except for any tags you edit security settings for individually. The user and group access settings require that both the local and networked computer have the same lookout.sec file installed. *Host Access* controls whether a particular computer can access data on your computer, no matter who is logged on that computer.

1. Open the Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**).

2. Select **Configure»Security**.

3. Click the **Data Access** tab, and click the **Edit** button. The **Access Rights** dialog box appears.

4. Configure the network security for user or group access. The large field in the center of this dialog box lists the groups and user accounts that have permission to work with the .scf file shown at the top of the dialog box.

   - To remove a user or group, select it and click the **Remove** button.

   - To change a user or group permission, select it and select from the options in the **Access** listbox.

   - To add a new user or group, click the **Add** button. The **Add Users and Groups** dialog box appears. Highlight the user or group you want to add and click the **Add** button. Set the **Access rights**. Click **Help** in the **Access Rights** dialog box for more information. Click **OK** when you are finished.

5.  Configure the network security for Host Access. Click the **Configure Host Access** button in the **Host and Tag Data Access Rights** dialog box. In the **Configure Network Access** dialog box, allow or deny access for specific networked computers.

    •   You can use the asterisk wildcard to enable or disable all computers or all computers in a set. For instance, entering `*.yourdomain.com` would select all the computers with that domain address. This is the same syntax used by the Server TCP/IP Access VI, available by selecting **Tools»Options**, then selecting **VI Server:TCP/IP Access**. Refer to the LabVIEW documentation for more information about the VI Server and wildcards you can use in the **TCP/IP Access List**.

    •   You can browse the network and select individual computers by clicking the **Browse** button. The **Network Browser** dialog box appears. Click the network node to expand the network view, and select the computer you want to allow or disable access for. Click the **OK** button to add this computer to the access list.

6.  If you have enabled BridgeVIEW 2.*x* networking in the Tag Configuration Editor (by selecting **Configure»Network» BridgeVIEW 2.x Networking»Allow Network Access**), it enables the **Advanced** button on the **Configure Network Access** dialog box. Clicking this button opens the **Network Settings** dialog box that functions in the same way it did in BridgeVIEW 2.*x*.

7.  Click the **OK** button.

## Setting a Proxy User Account

The proxy user is the account used for unrecognized access to your data. For example, if you set the proxy user as Guest (default), then any unidentified client who attempts to access your data will be given the access rights you have assigned to the Guest account. Follow these steps to set proxy user access:

1.  Select **Tools»Datalogging & Supervisory Control»Options** and click the **Advanced** tab.

2.  Click **Proxy User**.

3.  Specify the user name and password to use for the proxy user. The default setting for the proxy user is the built-in Guest account, which has no password unless you add one.

4.  Click **OK**.

**Note**   If you later change the password for the user account specified as the proxy user, you must change the password in the Set Proxy User dialog box as well.

## Setting an Engine User Account

You can specify an *engine user* to ensure that your local Tag Engine has access to network tag data no matter who is logged in to the local LabVIEW DSC module application. If a locally-defined tag in the Tag Engine attempts to access tag data across the network using Logos, the local Tag Engine uses the engine user account. In this case, the tag on the local computer was created with the **Server Name** set to **Logos** in the Tag Configuration Editor.

If the remote machine recognizes that account (if it is defined in its `lookout.sec` file), it grants the access rights defined for that account. If the remote machine does not recognize that account (it is not defined in its `lookout.sec` file), it grants the proxy user access rights that are defined on the remote machine.

Follow these steps to set up an engine user account.

1. Select **Tools»Datalogging & Supervisory Control»Options** and click the **Advanced** tab.

2. Click **Engine User**.

3. Specify the user name and password to use for the engine user. The default setting for the engine user is the built-in Administrator account.

4. Click **OK**.

**Note**   If you later change the password for the user account specified as the engine user, you must change the password in the Set Engine User dialog box as well.

## Setting Tag Configuration Editor Access

Click the **Tag Editor Access** tab to set who has access to the Tag Configuration Editor.

1. While logged in as Administrator, select **Tools»Datalogging & Supervisory Control»Options**.

2. Click the **Advanced** tab, then click the **Tools Access** button.

3. On the **Tag Configuration Editor** tab, and click the **Edit** button. The **Access Rights** dialog box appears.

4.  The large field in the center of this dialog box lists the groups and user accounts that have permission to work with the `.scf` file shown at the top of the dialog box.

    •   To remove a user or group, select it and click the **Remove** button.

    •   To change a user or group permission, select it and select from the options in the **Access** listbox.

    •   To add a new user or group, click the **Add** button. The **Add Users and Groups** dialog box appears. Highlight the user or group you want to add and click the **Add** button. Set the **Access rights**. Click **Help** in the **Access Rights** dialog box for more information.

5.  Click **OK** when you are finished.

## Setting Startup Login Options

You can set several login options, such as whether the last user is logged in automatically when LabVIEW starts or whether a dialog box appears at startup so you must log in manually.

1.  Log in as an administrator (**Tools»User Name**).

2.  Select **Tools»Datalogging & Supervisory Control»Options.**

3.  Click the **Advanced** tab.

4.  Click the **Security Preferences** button.

5.  Click the **Startup** tab.

6.  Set the login option to use when LabVIEW starts.

7.  Click **OK** twice.

## Checking User Privileges

Use the DSC Check Operator Privilege VI located on the **Functions»System»Security** palette. This VI checks the privileges of the current user to determine whether the user has a particular privilege.

## Disabling Special Keys

Complete the following steps to prevent anyone logged in below a certain security level from using the special key combinations, including <Ctrl-Alt-Delete>, <Ctrl-Esc>, <Alt-Esc>, <Alt-Tab>, <Alt-Enter>, <Ctrl-Alt-Esc>, and the Windows logo key.

1.  Log in as an administrator (**Tools»User Name**).

2.  Select **Tools»Datalogging & Supervisory Control»Options**.

3.  Click the **Advanced** tab.

4. Click the **Security Preferences** button.

5. Click the **Advanced** tab.

6. Set the security level at which you want to disable special keys.

7. Click **OK**.

# Prompting an Operator to Log into an Application

Use the NI SEC Invoke Login DialogVI located on the
**Functions»System»Security** palette. This VI launches the **Login** dialog
box and returns the user name and access level. You can have the
application control login as part of its HMI.

# Logging an Operator into or out of an Application Programmatically

Use the NI SEC Programmatic Login VI located on the **Functions»
System»Security** palette. To use this VI, you must enter a user name and
password. If successful, the user is logged in to the system, and no **Login**
dialog box appears.

Use the NI SEC Programmatic Logout VI located on the **Functions»
System»Security** palette. This VI logs the current user out of the
LabVIEW DSC module.

# Identifying the Current Operator

Use the NI SEC Interactive User Info VI located on the **Functions»
System»Security** palette to return the name and current operator name and
information. Use the NI SEC User Account Info VI to retrieve information
about any user account.

# 9

# Networking and Deploying Applications

This chapter describes how to set up applications for use on a network and how to deploy applications on computer systems and networks other than the system on which it was developed.

## Deploying Application Files

If you plan to deploy your application to one or more client machines, National Instruments recommends using both the Application Builder and LabVIEW DSC module Run-Time System to simplify the process. Contact National Instruments to order these software tools, if you do not already have them.

### LabVIEW DSC Run-Time System

You can deploy LabVIEW Datalogging and Supervisory Control (DSC) module applications using the LabVIEW DSC module Run-Time System, which has built-in support for the special LabVIEW DSC module capabilities. You must purchase the LabVIEW DSC module Run-Time System separately. You cannot run LabVIEW DSC module applications using just the LabVIEW Run-Time Engine.

When you deploy applications, make sure all the proper hardware and hardware drivers are installed and working in the server and client computers.

### Building an Executable Application from a VI

You can use the LabVIEW Application Builder to create applications from VIs that use the LabVIEW Datalogging and Supervisory Control (DSC) module. When you install the Application Builder, the **Build Application** selection appears in the **Tools** menu.

If your application does not use any LabVIEW DSC module features, you may use the Application Builder as you would with any LabVIEW application. If your application takes advantage of the features the LabVIEW DSC module adds to LabVIEW, additional software and files will be necessary for it to function correctly.

## Applications that Include LabVIEW DSC Module VIs

If your application uses VIs specific to the LabVIEW DSC module, you must install either a LabVIEW DSC module development package or the LabVIEW DSC module Run-Time System on the computers that will run the application. The LabVIEW DSC module Run-Time System correctly installs the LabVIEW DSC module VIs as well as all of the configuration tools, Tag Engine, historical datalogging, network data access, security, and other features provided by the LabVIEW DSC module. The Application Builder excludes these items specific to the LabVIEW DSC module from the built application, because they must be correctly installed on the target computer.

You must put your application in the same directory where the LabVIEW DSC module Run-Time System is installed. This directory will contain `dscrts.exe`. Run this executable if you did not build your own. If you used Application Builder to create your own executable, you do not need to run `dscrts.exe`.

## Applications that Include Only Security VIs

If the only LabVIEW DSC module VIs your application uses are the Security VIs, you can distribute those VIs without using the LabVIEW DSC module Run-Time System, as follows.

1.  Install your application on the deployment computer.

2.  Create a subdirectory on the deployment computer called `vi.lib`, in the same directory where you installed your application.

3.  Copy the following files from your LabVIEW DSC module development machine to `vi.lib` on the computer that will run the application, as shown in the following table.

**Table 9-1.** Files to Deploy for Security VIs Only

| Directory\File on LabVIEW DSC Module Computer | Destination Directory on Deployment Computer | What to Copy |
|---|---|---|
| `lv_init.vi` | `vi.lib` | File |
| `extensions\_core\*` | `vi.lib\extensions` | Entire subdirectory |
| `extensions\LVSecurity.vi` | `vi.lib\extensions` | File |
| `extensions\_security_stubs.llb` | `vi.lib\extensions` | File |
| `extensions\security\*` | `vi.lib\extensions` | Entire subdirectory |

4. To include the **Security** submenu in your application's **Tools** menu, create a `project` directory in your application's main directory. Then, copy the `Security` folder from the `LabVIEW\project\lvdsc\` directory into your application's `project` directory.

## What to Include with Your Application

You should include all VIs you have written for your application, as well as any external data files. For LabVIEW DSC module-based applications, these files will commonly include the following:

- Tag configuration (`.scf`) files

- Preference files (`.ini`, `.cfg`) from the LabVIEW directory

- Hardware configuration files (such as `.iak` files for FieldPoint, MAX configuration data, and so on)

- Common configuration database file (`.ccdb`). This file can be identified by examining the title bar of the Server Browser utility (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**), or by examining the following value in the Windows Registry key:
  `HKEY_LOCAL_MACHINE\Software\National Instruments\NI-Servers`
  `Value: Active CCDB`

- Server software for all servers your application depends on for data. You may have to register VI-based servers yourself.

These files may already be included, if you used Application Builder and included them at that time. For more information about files to include when you deploy your application, refer to the NI Developer Zone at `ni.com/zone`.

# Setting up Networked Applications

To use a LabVIEW DSC module application on a network, you need to register the computers or devices on your network that use the Logos networking protocol, make sure the clocks of all your computers or devices are synchronized, make sure your security files are compatible on your networked computers, verify network paths and names, and make sure you have all necessary services running. BridgeVIEW 2.*x* networking is available, though not recommended, as explained in the *Importing Network Tags* section of Chapter 4, *Using Tags to Manage I/O in LabVIEW*.

## Logos Networking Technology

National Instruments uses a special networking technology known as the Logos networking protocol, installed as a service on your computer when you install the LabVIEW DSC module. The Logos networking protocol functions across the network without you performing any special configuration or work; you can browse to any Logos data on your network from a software client with Logos capability (such as the Tag Configuration Editor or the Tag Configuration Wizard).

**Note**  LabVIEW without the DSC module can acquire Logos data through DataSocket, but cannot act as a Logos server. The LabVIEW DSC module allows you to connect tags and data directly through the Logos networking protocol and can act as a Logos server.

The LabVIEW DSC module also adds OPC functionality to LabVIEW, allowing LabVIEW DSC module tags to connect to OPC servers. Again, you do not need to perform any special configuration operations to use this OPC capability. The Tag Configuration Editor and the Tag Configuration Wizard can browse for any OPC servers on the network and access those tags.

## Registering Networked Computers

To access LabVIEW and Lookout applications or FieldPoint hardware using the Logos networking protocol, the FieldPoint device or the computer running an application must be registered. You can register computers through the Tag Configuration Editor, the Tag Monitor, the LabVIEW DSC module options, or the Server Browser.

**Note**  Some computers, particularly on a local subnet, automatically appear as registered computers.

Complete the following steps to register or unregister a computer.

1. Log in as Administrator or with administrator privileges (**Tools»User Name**).

2. Select **Tools»Datalogging & Supervisory Control»Options**, click the **Advanced** tab, and click **Registered Computers**. Optionally, you can open one of the following utilities:

   - Tag Configuration Editor (**Tools»Datalogging & Supervisory Control»Configure Tags**), then select **Configure»Network»Registered Computers**.

   - Tag Monitor (**Tools»Datalogging & Supervisory Control»Monitor Tags**), then right-click the **Network Neighborhood** node, and select **Register Computer** from the shortcut menu.

   - Server Browser (**Tools»Datalogging & Supervisory Control»Advanced»Server Browser**), then click the **Register Computer** button.

3. Complete the following steps to add a computer to the list of registered computers.

   a. Click the **Add** button in the **Registered Machines** dialog box. The **Register Computer** dialog box appears.

   b. Enter the computer you want to access in the **Computer Name** field, or browse for the computer in the network tree.

   c. Click the **Register** button, then click **OK** when you are finished.

4. Complete the following steps to remove a computer from the list of registered computers.

   a. In the **Registered Machines** dialog box, select the computer you want to remove.

   b. Click the **Remove** button, then click **OK**.

You also can unregister a computer in the Tag Monitor by right-clicking it and selecting **Unregister** from the shortcut menu.

## Setting up Time Synchronization for Networked Computers

To keep data properly time stamped, make sure the times on your computers are properly synchronized. The National Instruments time synchronization service is installed as a service in Windows 2000/NT/XP that runs every time you run your computer. Time synchronization runs as a background process in Windows Me/9*x*.

## Determining Time Server Search Order

Suppose you have four computers you need to have synchronized. You should choose a primary time synchronization server and backup(s). Make sure that the order of search for time servers is the same for all the computers on your network you want to synchronize, including the primary time synchronization server. If a time server fails, the other computers synchronize to the next one in line.

✎ **Note**  If you have both Windows 2000/NT/XP and Windows Me/9*x* computers on the same network, you might have better results if a Windows 2000/NT/XP computer is the primary time synchronization server.

Suppose you have computers A, B, C and D, where A is the primary time synchronization computer, B is the time synchronization computer if A fails, and so on. In this scenario you would use the time server search order shown in Table 9-2.

**Table 9-2.**  Time Synchronization Order

| Computer A | Computer B | Computer C | Computer D |
|:----------:|:----------:|:----------:|:----------:|
| None listed | A | A | A |
| — | — | B | B |
| — | — | — | C |

As the primary time server, Computer A would have no other servers listed. If Computer A is running, it should synchronize to itself. Computer B should synchronize to Computer A, if A is running. If A is not running, B should synchronize to itself. Computer C should synchronize to Computer A if it is running, Computer B if A is not running, and to itself if neither A nor B is running. This pattern should be used for all the computers you want in one synchronized set.

## Configuring Time Synchronization

1.  Select **Tools»Datalogging & Supervisory Control»Options**.
2.  Click the **Advanced** tab.
3.  Click the **Time Synchronization** button. The **Time Server Search Order** dialog box appears.

Any computer that is running the time synchronization service can serve as a time server or a time client. The primary time server is the first computer listed in the **Time Server Search Order** field. If no computer is set as a primary time server, your computer synchronizes to itself.

4. To add a computer to the **Time Server Search Order** field, click the **Add** button. If you know the name of the computer you want to add, you can type it into the **Computer name** field. If you do not know the exact name of the computer, you can browse for it in the network tree contained in the **Select a Computer** field.

   To remove a computer from the **Time Server Search Order** field, highlight the computer name and click the **Remove** button.

5. To change the order in which your computers search for a time synchronization server, select the computer name and click the **Up** or **Down** buttons.

   • If you have some computers running Windows Me/9*x* and other computers running Windows 2000/NT/XP in your network, you should list Windows 2000/NT/XP computers first in the server search list. Time synchronization works better between Windows Me/9*x* and Windows 2000/NT/XP systems when the Windows 2000/NT/XP computer is the server.

   • You do not need to include a computer running LabVIEW in its own list of time synchronization services.

6. Use **Sleep Time (seconds)** to set how long each computer waits between each synchronization. You should set the primary time synchronization server sleep time to 60 seconds.

   If the primary server is off-line for some reason, a computer scheduled to synchronize automatically seeks out the second computer on the synchronization server list. At the time of the next synchronization, the computer first looks for the primary server before seeking a secondary synchronization server.

7. Click the **OK** button.

8. Repeat steps 1 through 7 for all computers on your network that you want to synchronize to make sure that the order of search for time servers is the same for all the computers, including the primary time synchronization server.

## Duplicating Security Files for Networked Computers

For user accounts to work consistently across your network, you must use the same `lookout.sec` file for all your installed copies of Lookout or the LabVIEW DSC module. After you create the `lookout.sec` file, make a copy of it and place the copy into the Windows system directory of each of the other computers with which you want to be able to use these user accounts.

## Preserving Network Paths in Deployed Applications

LabVIEW DSC module applications use computer network names to log data and access data sources. When you run applications on a computer system different from your development system, either make sure that all the names and paths on your deployment system are identical to the names and paths on your development system, or make the necessary alterations in your applications to match the network names and data paths of your deployed system.

# Monitoring NI Services

The National Instruments Logos networking protocol requires three background services that run in Windows outside of any National Instruments applications. These services are known as Citadel Server, Classified Ads, and Time Synchronization. In the Windows 2000/NT/XP task manager, these services appear as `lkcitdl.exe`, `lkads.exe`, and `lktsrv.exe`. Under Windows 2000/NT/XP, these services run automatically. If you need to interact with these services, you can use the Services utility, found in **Start»Settings»Control Panel»Services**.



The LabVIEW DSC module installs a National Instruments services manager, denoted by an icon located in the system tray of the Windows taskbar, near the computer clock. The columns of circles from left to right represent Classified Ads, Time Server, and Citadel Service. A green light indicates the service is running. A red light indicates the service is stopped. When you right-click this icon, a shortcut menu appears from which you can start or stop any of the NI services.

⚠ **Caution**    Do *not* stop these services while the LabVIEW DSC module, the Tag Engine, or Lookout is running.

# Viewing Client Connections

To see what computers are currently accessing data from a LabVIEW DSC module application, use the Engine Manager. For more information, refer to the *Viewing Tag Engine Status* section in Chapter 4, *Using Tags to Manage I/O in LabVIEW*.

# Troubleshooting Communication Problems

If you encounter communication problems over a network, the following points may be of help to you.

- Make sure each networked machine can access the others across the network.

- Because the protocol used for network communication between networked Tag Engines is based on TCP/IP, make sure each computer has TCP/IP configured correctly on it.

- Each machine must have a unique IP address and a host name assigned to it. TCP/IP utilities such as ping (all operating systems) and nslookup (only Windows 2000/NT/XP) can be used to verify the address and the host name.

💡 **Tip**  Execute ping /? from a command prompt to access ping command help. Execute nslookup <Enter>, then ? at the prompt to access nslookup help.

- If your computers are separated by a firewall, some adjustments will be required. For more information about networking across firewalls, refer to the Developer Zone's Resource Library at ni.com/zone.

- Use Tag Monitor to monitor communication and determine whether data is accessible via Logos.

# Configuring Startup VIs

Startup VIs are VIs that run automatically when LabVIEW starts.

1. Select **Tools»Datalogging & Supervisory Control»Advanced» Startup VIs** to display the **Configure Startup VIs** window.

2. Click the **Add** button.

3. Navigate to the VI you want to run when LabVIEW starts and click the **Open** button.

4.  Place or remove checkmarks from the **Show Panel** and **Run** checkboxes.

    If you remove the checkmark from the **Show Panel** checkbox, the VI must open a reference to itself to continue running after it is loaded. Refer to the LabVIEW documentation for information about using the VI Server to open a reference.

5.  Select a VI in the **Startup VIs** listbox and click the **Move Up** or **Move Down** buttons to change the order in which the VI loads.

6.  Click the **OK** button.

# A

# Using SQL to Access Historical Data in Citadel

This chapter describes Structured Query Language (SQL), Open Database Connectivity (ODBC), and accessing Citadel data using both SQL and ODBC.

## Introduction

The Citadel historical database includes an ODBC driver, which enables other applications to directly retrieve data from Citadel using SQL queries.

### What is ODBC?

ODBC is a standard developed by Microsoft. It defines the mechanisms for accessing data residing in database management systems (DBMSs). Nearly all Windows applications that can retrieve data from a database support ODBC.

### What is SQL?

SQL is an industry-standard language used for retrieving, updating, and managing data. In LabVIEW with the Enterprise Connectivity toolkit and in Lookout, you can use SQL to build queries to extract data from Citadel. The Citadel ODBC driver also includes many built-in data transforms to simplify statistical analysis of retrieved data.

## Creating a Citadel ODBC Data Source

Complete the following steps to create a Citadel ODBC data source for use with the LabVIEW Datalogging and Supervisory Control (DSC) module.

1.  Click the Windows **Start** button and select **Settings»Control Panel**.

2.  Run the ODBC applet. It might be called **ODBC** or **ODBC Data Sources** or something similar, depending on your operating system version number.

**Note**   Shut down all ODBC applications, such as databases, spreadsheets, word processors, and Microsoft Query, before you run the ODBC applet.
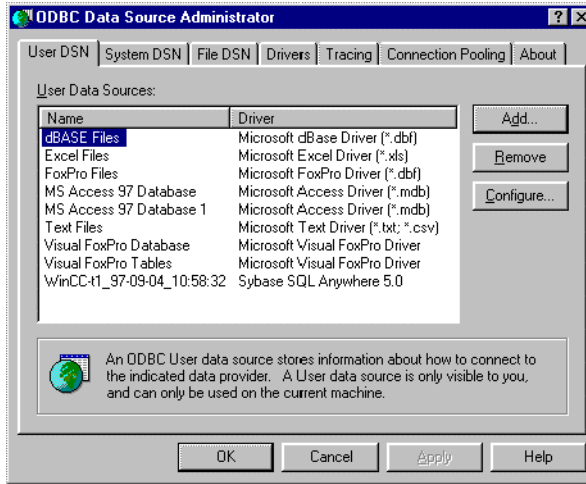


3.  Click the **User DSN** tab or the **System DSN** tab, depending on which type of data source you want to create. User DSNs are only visible to the user who created them on the current machine. System DSNs are available to all users on the current machine.

4.  Click the **Add** button. The following dialog box appears.
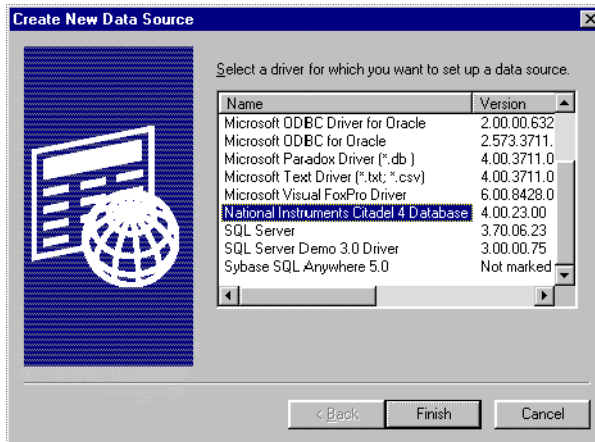
5.  Select **National Instruments Citadel 4 Database**, then click **Finish**.

6.  In the **National Instruments Citadel ODBC Setup** dialog box, fill in the **Data Source Name**, **Description**, and **Database Path** fields.

    a.  The **Data Source Name** is the name that ODBC applications use to select the data source.

    b.  **Description** is a free-form text string you can enter to describe the data source.

    c.  **Database Path** should match the location of the Citadel database you intend to access. Use a fully qualified path to a remote computer if you are accessing the Citadel database on a remote computer, such as `\\tripper\c:\data`.



The **Data Source Name** must be different from any other ODBC data source name. The **Description** is arbitrary. The **Database Path** gives the location of the folder where the data for this source are stored. If your `.scf` files use different data locations, you probably want to configure one ODBC data source for each.

7.  Click the **OK** button in the **Setup** dialog box, then click the **OK** button in the **ODBC Data Source Administrator** dialog box.

**Note**   Some applications are not completely ODBC compliant. If you plan to use Microsoft Query, Microsoft Access, or Visual Basic, make sure **Maximum Column Name Length** does not exceed 62 characters. These applications cannot handle longer names. Applications that are completely ODBC compliant can handle names up to 126 characters

long. All traces whose names exceed the **Maximum Column Name Length** are excluded from queries.

Because tags might include network path information in their names, you might exceed the 62 and 126 character limitations of ODBC. Consideration given to naming and organizing objects can minimize the risk of encountering this difficulty.

If you plan to use Microsoft Access or Visual Basic, select **Convert special characters** to force your tag names into an accepted format by replacing characters within the names with the characters in Table A-1.

**Table A-1.**  Special Access SQL Characters

| Special Character | Converted Character |
| :---: | :---: |
| period (.) | at sign (@) |

The special characters changed in ODBC 4. If you are converting SQL queries from an earlier version earlier of the ODBC driver, you might have to rewrite any SQL queries you set up in your earlier processes.

# Accessing Citadel Data

## Traces Table

The ODBC driver presents Citadel data to other applications as a traces table. The table contains a field or column for each tag logged to the Citadel database and three fields you can use to specify query criteria and to time stamp retrieved data: **Interval**, **LocalTime**, and **UTCTime**.

**Interval** specifies the query value sample rate. **Interval** can range from 10 ms to several years. **Interval** defaults to 1 (one day). **WEEK** is a standard seven days, but **MONTH** and **YEAR** account for different month lengths and leap years.

Because Citadel is event-driven, it only logs a value when the value changes. Using **Interval**, you can query Citadel for values evenly spaced over a period of time.

**LocalTime** and **UTCTime** are time-stamps that indicate when values are logged. Citadel stores the time in **UTCTime** format and derives **LocalTime** from the stored time.

The following `where clause` query uses **Interval** and **LocalTime** to select data over a specified time at one-minute intervals.

```
SELECT * FROM Traces
WHERE LocalTime>"12/1 10:00"
   AND LocalTime<"12/2 13:00"
   AND Interval="1:00"
```

# Points Table

The Points table is used to retrieve the actual values logged for a tag and the times they were logged. Because logging to Citadel takes place asynchronously, there is no correlation between the timestamps for one tag and another. For this reason, when querying the Points table, you can query only one tag at a time.

The *where* clause using **LocalTime** and **UTCTime** is supported for the Points table; however, **Interval** is not relevant to the Points table. The data transforms are also not relevant to the Points table and are not supported.

An example of a query using the Points table could be
```
SELECT LocalTime, "\\computername\process\Pot1" FROM
Points
WHERE LocalTime > "12/1 10:00" AND
                  LocalTime < "12/2 10:00"
```

# Data Transforms

Queries can include special commands that perform data transforms to manipulate and analyze historical data. Table A-2 lists data transform commands.

**Table A-2.** Data Transform Commands

| Command | Transformation |
|---------|----------------|
| Min{Datapoint} | Returns the minimum for *Datapoint* across the interval. |
| Max{Datapoint} | Returns the maximum for *Datapoint* across the interval. |
| Avg{Datapoint} | Returns the average for *Datapoint* across the interval. |
| Stdev{Datapoint} | Returns the standard deviation for *Datapoint* across the interval. |
| Starts{Datapoint} | Returns the number of starts (that is, the number of transitions from OFF to ON) for *Datapoint* across the interval. For numeric points, 0.0 is interpreted as OFF, and all other numbers are treated as ON. |

**Table A-2.** Data Transform Commands (Continued)

| Command | Transformation |
|---------|----------------|
| Stops{Datapoint} | Returns the number of stops (that is, the number of transitions from ON to OFF) for *Datapoint* across the interval. |
| ETM{Datapoint} | Returns the amount of time *Datapoint* was in the ON state across the interval. |
| Qual{Datapoint} | There might be gaps in the historical data traces in Citadel because of machine shutdown, Tag Engine shutdown, or a similar occurrences. Qual returns the ratio of time for which valid data exist for *Datapoint* across the interval to the length of the interval itself. If valid data exist for only one-half of the interval, Qual returns 0.5. |

Using these data transforms, you can directly calculate and retrieve complex information from the database such as averages and standard deviations, so you do not need to extract raw data and then manipulate them in another application.

For example, you need to know how many times a compressor motor started in December. You also need to know its total run time for the month. Use the following query to get answers:

```
SELECT
"Starts{\\computername\processname\PLC.MotorRun}",
      "ETM{\\computername\processname\PLC.MotorRun}"
FROM Traces
WHERE LocalTime>="12/1/95"
  AND LocalTime<"1/1/96"
  AND Interval="31"
```

## SQL Examples

The following examples are typical query statements; however, queries might be much more involved, depending on your system requirements.

```
SELECT *
FROM Traces
```

Retrieves the current value of every data member logged to Citadel. Because your query does not occur at the same moment in time as a PLC poll, signals scanned from PLCs are not included in the
```
retrieved data.
```

```
SELECT *
FROM Traces
WHERE Interval="0:01"
```

Retrieves the value of every tag logged today in one-second increments. Notice that the interval value is enclosed in quotation marks.

```
SELECT LocalTime, "\\computername\processname\Pot1"
FROM Traces
WHERE LocalTime>"12/2/2000 16:50"
AND Interval="0:01"
```

Retrieves and time stamps the value of Pot1 in one-second increments from 4:50 p.m. to now. Names are enclosed by quotes.

```
SELECT LocalTime, "\\computername\processname\AB1.I:3",
"Max{\\computername\processname\AB1.I:3}"
FROM Traces
WHERE LocalTime>"10/1/95"
AND LocalTime<"11/1/95"
AND Interval="1:00"
```

Retrieves and time stamps an Allen-Bradley PLC input in one-minute intervals for the month of October. This query also indicates the highest occurring input value of each minute.

```
SELECT LocalTime,
"\\computername\processname\OVEN1_SP",
"\\computername\processname\PLC.OVEN1_PV",
"Max{\\computername\processname\PLC.OVEN1_PV}",
"Min{\\computername\processname\PLC.OVEN1_PV}",
"Avg{\\computername\processname\PLC.OVEN1_PV}"
FROM Traces
WHERE LocalTime>"12/2/2000 14:00"
   AND LocalTime<="12/2/2000 17:00"
   AND Interval="1:00:00"
```

Retrieves an oven temperature at 3:00 p.m., 4:00 p.m., and 5:00 p.m. and shows the highest, lowest, and average temperatures for every hour between 2:00 p.m. and 3:00 p.m., 3:00 p.m. and 4:00 p.m., and 4:00 p.m. and 5:00 p.m.

## Accessing Citadel Data with Microsoft Query

1. Launch Microsoft Query.

2. Select **File»New**. If the Citadel data source is not visible, you must create it. If it is visible, skip to step 3.

   a. Select the <New Data Source> entry in the **Databases** tab. Make sure the **Use the Query Wizard to create/edit queries** option is *not* selected.
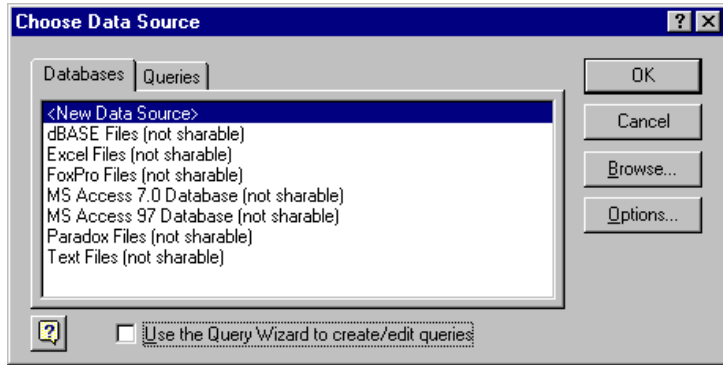
   

   b. The **Create New Data Source** dialog box appears. Fill in the fields as shown in the following illustration.
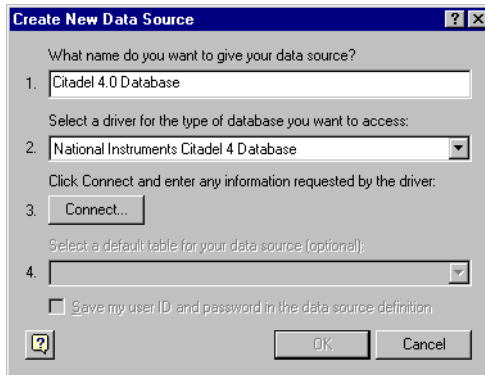
   

   c. Click the **Connect** button. The **National Instruments Citadel ODBC Setup** dialog box appears. Fill in the fields as shown in the following illustration. Enter the location of the database in the **Database Path** field.
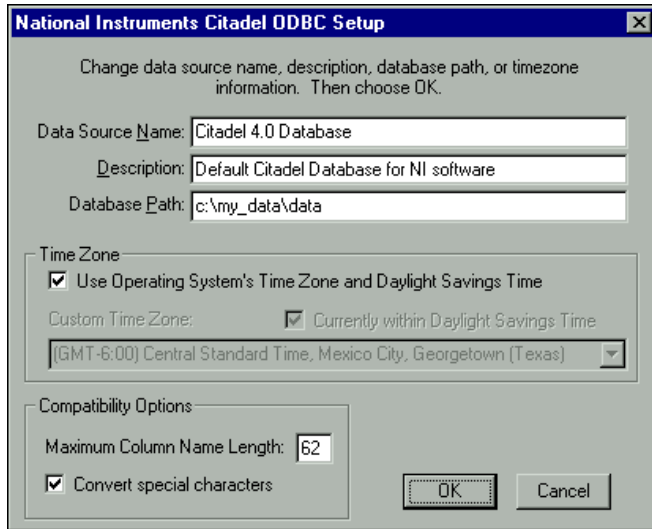
d. Click the **OK** button. The **Create New Data Source** dialog box reappears, and should look like the following illustration.
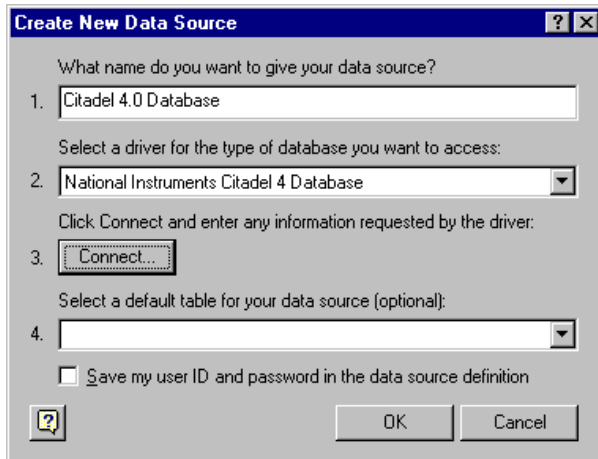


e. Do *not* select a default table or save your ID and password. Click the **OK** button. The **Choose Data Source** dialog box appears, this time with the Citadel database as one of the data source choices. Select Citadel as the data source and click the **OK** button.
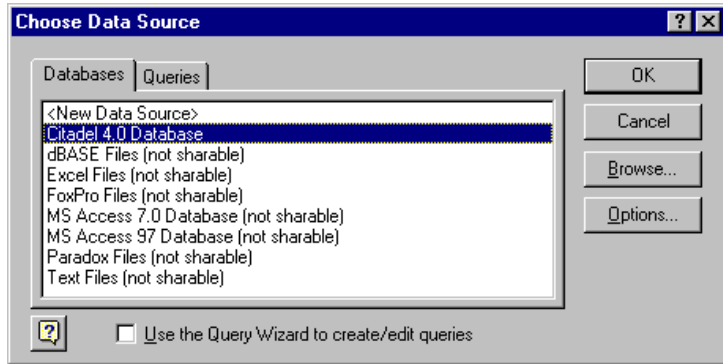
3. Select the Citadel data source. Make sure the **Use the Query Wizard to create/edit queries** option is *not* selected.

**Note**  If Microsoft Query is unable to connect to Citadel, make sure you have logged data to Citadel and entered the correct database path in the **ODBC Setup** dialog box.

**Note**  If Citadel is not listed in the **Data Source** dialog box, make sure you have created Citadel as a data source. Refer to the *Creating a Citadel ODBC Data Source* section for more information.

4. In the **Add Tables** dialog box, double-click **Traces** or **Points**, depending on whether you want to view raw data (**Points**) or resampled data (**Traces**). In the following figure, **Traces** are used.
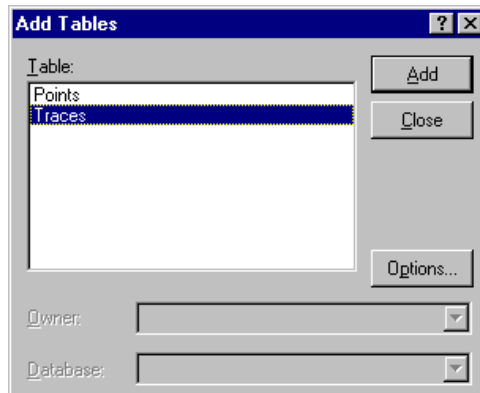
5. Close the dialog box.

   Microsoft Query presents the full Query Window with the `Traces` and `Points` tables. The names in these tables are a comprehensive list of all name values that have been logged to Citadel.
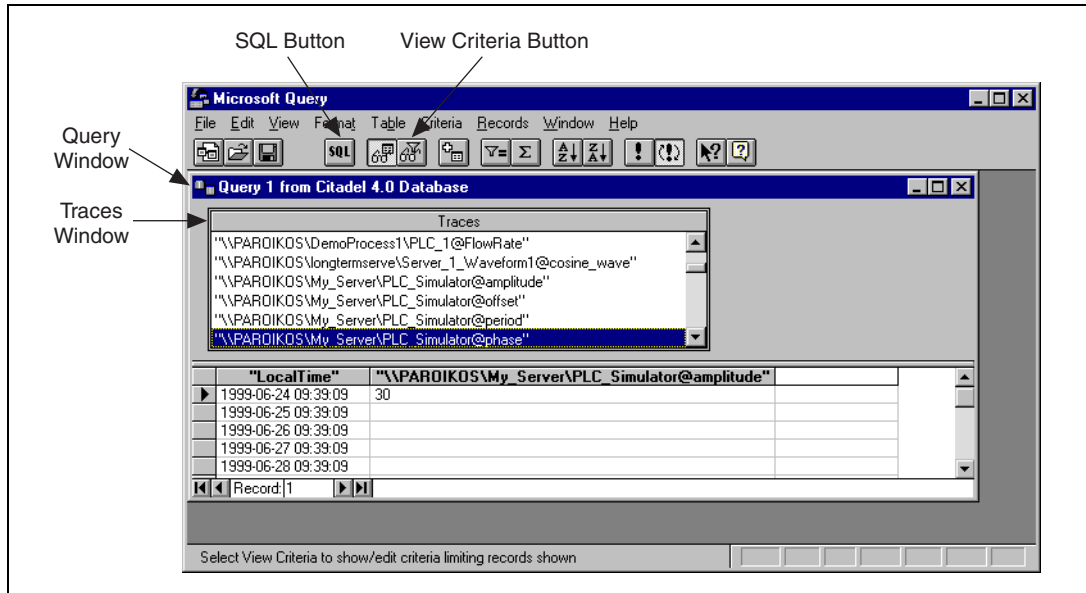


**Figure A-1.** The Query Window with Traces Table

6. To view a trace, double-click or drag the field you want to view to a blank column in the data pane. In Figure A-1, `LocalTime` and `"\\PAROIKOS\My_Server\PLC_Simulator@amplitude"` have been dragged down.

7. To view a data transform value, enter the function directly into a blank column. For example, to view the minimum value of `PLC_Simulator.amplitude`, you would enter `"min{\\PAROIKOS\My_Server\PLC_Simulator@amplitude}"`. You must include the quotation marks and braces.

   The data set in Figure A-1 was retrieved using no specific criteria, so the ODBC driver used the default. Although there are several ways to specify criteria, this example uses the criteria pane.

8. Click the **View Criteria** button. The pane appears in the **Query** window.

9.  Add a field to the criteria pane by double-clicking the field, or by dragging it to the blank column in the criteria pane. In the following example, an Interval criteria of one minute was chosen.
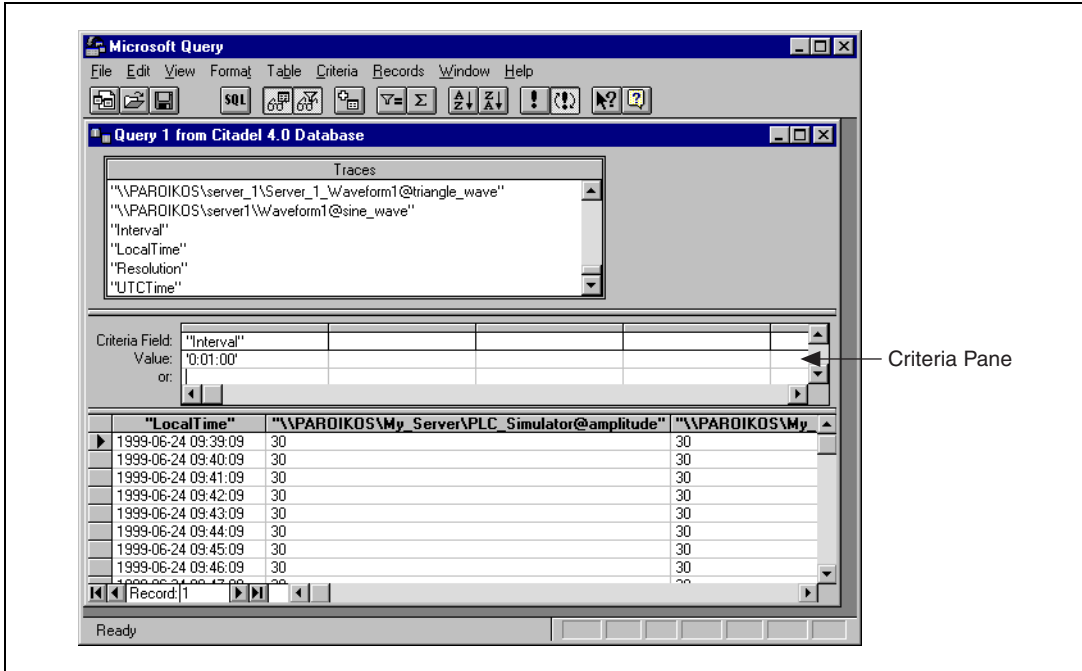


**Figure A-2.**  Query Window with Criteria Pane

When you enter qualifying criteria values, use the syntax demonstrated in *SQL Examples* earlier in this chapter. Refer to the following figure for more information.

As soon as you specify criteria, Microsoft Query retrieves the specified data. You can save the query at any stage of development, and as you build the query, the application builds an SQL statement.

10. Click the **SQL** button to view or edit the query statement.
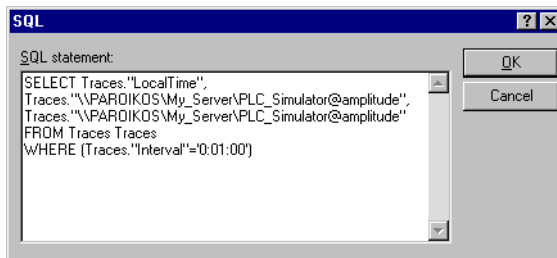
## Accessing Citadel Data from Other Software

Refer to National Instruments Developer Zone, at `ni.com/zone` for information about using Citadel with other software such as Microsoft Excel, Microsoft Access, or Visual Basic. Use a search phrase such as `Citadel SQL`.

# B

# Technical Support Resources

## Web Support

National Instruments Web support is your first stop for help in solving installation, configuration, and application problems and questions. Online problem-solving and diagnostic resources include frequently asked questions, knowledge bases, product-specific troubleshooting wizards, manuals, drivers, software updates, and more. Web support is available through the Technical Support section of ni.com.

## NI Developer Zone

The NI Developer Zone at ni.com/zone is the essential resource for building measurement and automation systems. At the NI Developer Zone, you can easily access the latest example programs, system configurators, tutorials, technical news, as well as a community of developers ready to share their own techniques.

## Customer Education

National Instruments provides a number of alternatives to satisfy your training needs, from self-paced tutorials, videos, and interactive CDs to instructor-led hands-on courses at locations around the world. Visit the Customer Education section of ni.com for online course schedules, syllabi, training centers, and class registration.

## System Integration

If you have time constraints, limited in-house technical resources, or other dilemmas, you may prefer to employ consulting or system integration services. You can rely on the expertise available through our worldwide network of Alliance Program members. To find out more about our Alliance system integration solutions, visit the System Integration section of ni.com.

# Worldwide Support

National Instruments has offices located around the world to help address your support needs. You can access our branch office Web sites from the Worldwide Offices section of `ni.com`. Branch office Web sites provide up-to-date contact information, support phone numbers, e-mail addresses, and current events.

If you have searched the technical support resources on our Web site and still cannot find the answers you need, contact your local office or National Instruments corporate. Phone numbers for our worldwide offices are listed at the front of this manual.

# Glossary

| Prefix | Meanings | Value |
|--------|----------|-------|
| m- | milli- | $10^{-3}$ |
| c- | centi- | $10^{-2}$ |

## A

A
amperes.

access level
Numeric value between 0 and 10 that can be used to control access to an HMI.

ACK (Acknowledge)
The sequence action that indicates recognition of a new alarm.

alarm
An abnormal process condition. In the LabVIEW DSC module, an alarm occurs if a tag value goes out of its defined alarm limits or if a tag has bad status.

Alarm Summary
A display of tags currently in alarm, or a display of tags previously in an unacknowledged alarm state that have returned to a normal state.

analog tag
A continuous value representation of a connection to a real-world I/O point or memory variable. This type of tag can vary continuously over a range of values within a signal range.

application
The application created using the LabVIEW Datalogging and Supervisory Control Module Development System and run in the LabVIEW Datalogging and Supervisory Control Module Runtime System environment.

ASCII
American Standard Code for Information Interchange.

## B

batch process
Process or production run that results in a batch of logged data, also known as a data set.

| | |
|---|---|
| bit array tag | A multibit value representation of a connection to a real-world I/O point or memory variable. In the LabVIEW DSC module, this type of tag can be comprised of up to 32 discrete values. |

# C

| | |
|---|---|
| Citadel | A database for storing historical tag values. |

# D

| | |
|---|---|
| DAQ | Data Acquisition. |
| DataSocket | Both a technology and a group of tools that facilitates the exchange of data and information between an application and a number of different data sources and targets. It provides one common API to a number of different communication protocols. |
| data set | A group of tag values logged together as a set during a specified period of time. |
| DDE | Microsoft Dynamic Data Exchange protocol. |
| deadband | In process instrumentation, the range through which an input signal can vary, upon reversal of direction, without initiating an observable change in output signal. Deadband is usually expressed in percent of range. *See also* log deadband and update deadband. |
| device server | An application that communicates with and manages a peripheral hardware device such as a Programmable Logic Control (PLC), remote I/O device or plug-in device. Device servers pass tag values to the Tag Engine in real time. |
| discrete tag | A two-state (on/off) value representation of a connection to a real-world I/O point. In the LabVIEW DSC module, this type of tag can be either a one (TRUE) or a zero (FALSE). |
| DSC | Datalogging and Supervisory Control. |
| dynamic attributes | Tag attributes that do not require the Tag Engine to be restarted when they are edited or reconfigured. Examples of dynamic attributes include enabling logging operations, alarm attributes, and some scaling attributes. *See also* static attributes. |

# E

| | |
|---|---|
| Engine | *See* Tag Engine. |
| engine user | You can specify an *engine user* to ensure that your local Tag Engine has access to network tag data no matter who is logged in to the local LabVIEW DSC module application. If a locally-defined tag in the Tag Engine attempts to access tag data across the network using Logos, the local Tag Engine uses the engine user account. |
| engineering units (EU) | Terms of data measurement, as degrees Celsius, pounds, grams and so on. |
| event | Something that happens to a tag. Events include tags going into or out of alarm state and the user setting a tag value. |

# G

| | |
|---|---|
| group | *See* tag group or I/O group. |

# H

| | |
|---|---|
| historical trend | A plot of data (values versus time) showing values that were logged to disk. |
| Historical Trend Viewer (HTV) | A utility that accesses historical data from the Citadel historical database. |
| HMI Wizard | A utility in the LabVIEW Datalogging and Supervisory Control module that automates the process of generating HMI block diagram code. |
| host access | *Host Access* controls whether a particular computer can access data on your computer, no matter who is logged on that computer. |
| Human Machine Interface (HMI) | A graphical user interface for the user to interact with the LabVIEW Datalogging and Supervisory Control module system. |

# I

| | |
|---|---|
| IA | Industrial Automation. |
| IAK server | Also known as DLL-based servers or Industrial Automation Servers (IAS). |

| | |
|---|---|
| ID tag | When logging data sets, the *ID tag* for each data set denotes a particular data set and the time during which the data set run took place. |
| input tag | A tag that accepts Real-Time Database values from a device server. |
| input/output (I/O) tag | A tag that accepts Real-Time Database values from a device server and sends values to the server. |
| IP | Internet Protocol. |
| I/O group | A set of related server items, all of which share the same server update rate and deadband. |
| item | A channel or variable in a real-world device that is monitored or controlled by a device server. |

# L

| | |
|---|---|
| LabVIEW Datalogging and Supervisory Control Module Run-Time System | An execution environment for applications created using the LabVIEW Datalogging and Supervisory Control Module Development System. |
| LabVIEW RT | LabVIEW Real-Time software. |
| log deadband | The range through which a tag value must change before it is logged to Citadel. |
| log resolution | The smallest change in a tag value stored in the historical database. |

# M

| | |
|---|---|
| m | meters. |
| Man Machine Interface (MMI) | *See* Human Machine Interface (HMI). |
| MAX | Measurement and Automation Explorer, a National Instruments configuration environment. |
| memory tag | A tag not connected to a real-world I/O point. Memory tags are used for user-defined calculations. *See also* tag and network tag. |

# N

| | |
|---|---|
| network tag | A tag remotely connected to any type of tag on another Tag Engine. *See also* tag and memory tag. |

# O

| | |
|---|---|
| ODBC | Open Database Connectivity. A standard developed by Microsoft, it defines the mechanisms for accessing data residing in database management systems |
| OPC | OLE for Process Control. A COM-based standard defined by the OPC Foundation that specifies how to interact with device servers. COM is a Microsoft 32-bit Windows technology. |
| operator | The person who initiates and monitors the operation of a process. |
| output tag | A tag that sends values to a device server when it is updated in the Real-Time Database. |

# P

| | |
|---|---|
| Panel Wizard | A utility in the LabVIEW Datalogging and Supervisory Control module that automates the process of creating front panel controls. |
| PID | *See* Proportional Integral Derivative (PID) Control. |
| PLC | *See* programmable logic controller (PLC). |
| polling | A method of periodically observing each I/O point or user interface control to determine if it is ready to receive data or request computer action. |
| programmable logic controller (PLC) | A device with multiple inputs and outputs that contains a program you can alter. LabVIEW DSC module device servers establish communication with PLCs. |
| Proportional Integral Derivative (PID) Control | A combination of proportional, integral, and derivative control actions. Refers to a control method in which the controller output is proportional to the error, its time history, and the rate at which it is changing. The error is the difference between the observed and desired values of a variable that is under control action. |

| | |
|---|---|
| proxy user | You can specify a *proxy user* account as a way to handle unrecognized access requests to your data. |

# R

| | |
|---|---|
| Real-Time Database (RTDB) | An in-memory snapshot of all tags in the system. |
| real-time trend | A plot of data (values versus time) that is updated as each new point is acquired in the Real-Time Database. |
| reentrant execution | Mode in which calls to multiple instances of a subVI can run in parallel with distinct and separate data storage. |
| RTDB | *See* Real-Time Database (RTDB). |

# S

| | |
|---|---|
| s | seconds. |
| sampling period | The time interval between observations in a periodic sampling control system. |
| SCADA | Supervisory Control and Data Acquisition. |
| scf | A .scf file is a configuration file that stores tag information and Tag Engine parameters. |
| SCXI | SCXI Signal Conditioning eXtensions for Instrumentation; the National Instruments product line for conditioning low-level signals within an external chassis near sensors. |
| sensor | A device that produces a voltage or current output representative of some physical property being measured, such as speed, temperature, or flow. |
| SQL | Structured Query Language. SQL is an industry-standard language used for retrieving, updating, and managing data. You can use SQL to build queries to extract data from Citadel. |
| static attributes | Tag attributes that require the Tag Engine to be restarted if they are edited or reconfigured. Examples of static attributes include general attributes and I/O connection attributes, such as server, device, or item. *See also* dynamic attributes. |

| | |
|---|---|
| string tag | An ASCII or binary character representation of a connection to a real-world I/O point. |
| supervisory control | Control in which the control loops operate independently subject to intermittent corrective action. |
| synchronized | To keep data properly time stamped, make sure the times on your computer clocks are properly synchronized. |
| system developer | The creator of the application to be run in the LabVIEW Datalogging and Supervisory Control Module Runtime System. |
| system errors | Errors that happen in the LabVIEW Datalogging and Supervisory Control module system, like a server going down. System errors are displayed in a dialog box, on the Engine User Interface, and also are logged in a `syslog` file. |
| system events | Events that occur in the LabVIEW DSC module, like an operator logging on or a utility starting up. System events are logged in a `syslog` file. |

# T

| | |
|---|---|
| tag | A connection to a real-world I/O point or a memory variable. Tags can be one of four data types: analog, binary, discrete, or string. |
| tag attributes | Parameters pertaining to a tag, like its alarm, limits, or engineering units. Tag attributes are configured in the Tag Configuration Editor but can be changed dynamically using the Tag Attributes VIs. |
| Tag Configuration Editor | A utility to configure various parameters of a tag, such as connection information, scaling, or logging. |
| Tag Engine | Maintains the Real-Time Database of all tag values and alarm states. The Tag Engine runs as a separate process, independent of the HMI application. |
| tag group | A set of tags primarily used for reporting and acknowledging alarms. A tag can be associated with only one tag group. All tags belong to the group `<ALL>` by default. |
| Tag Monitor | A utility to view the current value of a tag, along with its status and alarm state. |

| | |
|---|---|
| tag status | A value that describes the validity of a tag value. A negative status represents an error, a positive status represents a warning, and a status of zero represents a good tag value. |
| TCP/IP | Transmission Control Protocol on top of the Internet Protocol. Enables communication between different types of computers and computer networks. |
| timestamp | The exact time and date at which a tag value was sampled. Tag values are stored with their timestamps in the RTDB. |
| trend | A view of data over time. Trends can display real-time or historical data. |

# U

| | |
|---|---|
| update deadband | The range through which a tag value must change before it is updated in the Real-Time Database. |
| URL | Uniform Resource Locator. A URL is the way Logos networking protocol locates data. |

# V

| | |
|---|---|
| V | Volts. |
| VI | Virtual Instrument. Program in LabVIEW that models the appearance and function of a physical instrument. |

# W

| | |
|---|---|
| Wizard | *See* HMI Wizard *and* Panel Wizard. |
| Wizard lock | A glyph that appears on a tag loop to indicate the LabVIEW Datalogging and Supervisory Control module has protected the association between a front panel object and the automatically generated block diagram. If a Wizard lock exists on a tag loop, you cannot modify that block diagram. After you release the Wizard lock, the association is broken and the Wizard no longer protects that tag loop. |

# Index

## A

Acknowledge Alarms VI, 6-6

Administrator Account, 8-3

alarm & event display control, 6-3 to 6-5
  acknowledging alarms, 6-3
  filtering alarms, 6-4 to 6-5

Alarm and Events VI, 6-7

alarm deadbands, setting
  analog tags, 4-28 to 4-29
  example, 4-12
  overview, 4-12

alarm summary display
  acknowledging alarms, 6-6
  reporting and viewing alarm status, 6-7
  using, 6-5

alarms
  acknowledging
    alarm & event display control, 6-3
    alarm summary and event history
      displays, 6-6
    keeping alarms unacknowledged, 4-30
  archiving and extracting alarm
    information, 6-7
  definition, 6-1
  filtering in alarm and event display
    control, 6-4 to 6-5
  logging and printing, 6-2
  overview, 6-1
  setting, 4-27 to 4-30
    analog tags, 4-28
    bit array tags, 4-29 to 4-30
    discrete tags, 4-29
    keeping alarms unacknowledged, 4-30
    string tags, 4-30
    types of alarms, 4-28

viewing, 6-2 to 6-7
  alarm and event display control,
    6-3 to 6-5, 6-7
  alarm summary display, 6-5
  overview, 6-3

analog tags
  purpose and use, 4-14
  scaling, 4-23 to 4-26
    assigning units, 4-26
    linear scaling example, 4-25
    offset example values (table), 4-25
    procedure for, 4-23
    square root and linear scaling,
      4-23 to 4-26
      example values (table), 4-24
      square root example, 4-25
  setting alarm deadband, 4-28 to 4-29
  setting alarms, 4-28

application deployment
  applications including LabVIEW DSC
    Module VIs, 9-2
  applications including only security VIs,
    9-2 to 9-3
  building executable application from VIs,
    9-1 to 9-2
  LabVIEW DSC Run-Time System, 9-1
  preserving network paths, 9-8
  what to include with your application, 9-3

application development, 2-1 to 2-3. *See also*
  HMI development.
  basic steps, 2-2 to 2-3
  example applications, 2-1
  parts of LabVIEW DSC module
    applications, 2-1 to 2-2

Archive Database VI, 7-10

archiving and extracting alarm and event
  information, 6-7

# D

DAQ channels, virtual, importing as tags, 4-7
data access restrictions, setting, 8-7 to 8-10
    engine user account, 8-9
    examples of handling access attempts,
      8-7 to 8-8
    network access, 8-8 to 8-9
    proxy user accounts, 8-8 to 8-9
data set, defined, 7-3
data set logging, 7-3 to 7-9
    batch logging, 7-3
    considerations for data set logger, 7-9
    creating data set for logging, 7-3 to 7-9
      Data Set Logger Configuration
        dialog box, 7-3 to 7-4
      data set logger server items (table),
        7-7 to 7-8
      data set run start/end conditions
        (table), 7-5
    editing data set for logging, 7-9
    ID tag for data sets, 7-3
    retrieving logged data sets, 7-9
database. *See* Citadel Historical Database;
    Real-Time Database.
Datalogging & Supervisory Control palette
    view, 1-8
Datalogging and Supervisory Control
    module, 1-1
dcomcnfg.exe, for accessing remote OPC
    servers, 3-9 to 3-10
DDE servers
    configuring tag attributes, 4-17 to 4-18
    overview, 3-2
    registering, 3-3
    using DDE servers with LabVIEW DSC
      module, 3-11
deadbands for tags, 4-10 to 4-12
    alarm deadbands
      analog tags, 4-28 to 4-29
      setting, 4-12

    interaction of deadband settings,
      4-10 to 4-11
    I/O group deadbands, 4-11
    log deadbands, 4-11
    overview, 4-10
    update deadbands, 4-11
default values for tag configuration fields,
    defining, 4-9 to 4-10
deleting tags, 4-13
deploying applications. *See* application
    deployment.
developing applications. *See* application
    development; HMI development.
device names, configuring, 4-19
device resources, configuring, 4-19 to 4-20
device servers. *See also* servers.
    definition, 3-1
    developing device servers, 3-11 to 3-12
    IA device servers, 3-2
    unregistering, 3-4
disabling special keys, 8-11 to 8-12
discrete tags
    purpose and use, 4-14
    scaling, 4-26
    setting alarms, 4-29
displaying data. *See* HMI development.
DLL-based servers, 3-2
documentation
    conventions used in manual, *iv*
    related documentation, 1-1 to 1-2
DSC Check Operator Privilege VI, 8-11
DSC module, 1-1
dynamic attributes, 4-15

# E

editing data sets for logging, 7-9
editing tag configuration, 4-7 to 4-9
    manually, 4-7 to 4-8
    programmatically, 4-8

# H