LabVIEW™

Datalogging and Supervisory Control Module Run-Time System Manual

NATIONAL INSTRUMENTS™

April 2003 Edition
Part Number 321635F-01

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 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
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, Canada (Vancouver) 514 685 7530,
China 86 21 6555 7838, Czech Republic 420 2 2423 5774, Denmark 45 45 76 26 00,
Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427,
India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970,
Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466,
New Zealand 1800 300 800, Norway 47 0 66 90 76 60, Poland 48 0 22 3390 150, Portugal 351 210 311 210,
Russia 7 095 238 7139, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

© 1996–2003 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

Citadel™, DataSocket™, FieldPoint™, LabVIEW™, Lookout™, National Instruments™, NI™, NI Developer Zone™, and ni.com™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Conventions

The following conventions are used in this manual:

»	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 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
Alarms and Events

Chapter 5
Historical Datalogging and Extraction

Chapter 6
Security

Chapter 7
Networking and Running Applications

Appendix A
Using SQL to Access Historical Data in a Citadel Database

Appendix B
Technical Support and Professional Services

Glossary

Index

1 Introduction

The LabVIEW Datalogging and Supervisory Control (DSC) Module Run-Time System provides an environment to run applications developed with the DSC Module.

You cannot edit the VIs used to create an application with the DSC Module Run-Time System. The system consists of a set of VIs for the application and supporting LabVIEW functionality, the definition of all data points in the system (tags), and the configuration of the servers that provide data to LabVIEW and the application.

This manual does not describe the specific nature of the application you might be running. Instead, it describes the features of the DSC Module Run-Time System, its architecture, execution system, and configuration tools. The developer of your application might provide additional documentation for the application. Consult the application developer for specific questions about your application.

Related Documentation

Select **Start»Programs»National Instruments»LabVIEW DSC Run-Time Help** to access the *LabVIEW Datalogging and Supervisory Control Module Run-Time System Help*. You also can select **Help» DSC Module Run-Time Help** from your application.

The application developer might include online help specific to your application. Refer to your application documentation or check with the developer for more information about the availability of application-specific help.

Utilities

The following utilities are installed with the DSC Module Run-Time System. You can access many of these features from the **Tools** menu or from the DSC Module Run-Time System window shown in Figure 1-1.

Figure 1-1. DSC Module Run-Time System Window

Note Your application might not allow access to the **Tools** menu. If you require access to this menu, contact the application developer.

Tag Configuration Editor

Use the Tag Configuration Editor to create, edit, or delete all tags in the DSC Module system and to configure Tag Engine parameters. Select **Tools»DSC Module»Configure Tags** to open the Tag Configuration Editor from your application. You also can click the button shown at left on the DSC Module Run-Time System window.

Caution Editing or deleting tags might cause the application to function incorrectly. Before you make any changes to the configuration (`.scf`) file, contact the application developer.

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, time stamp, 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»DSC Module»Monitor Tags** to open the Tag Monitor from your application. You also can click the button shown at left on the DSC Module Run-Time System window.

Tag Engine

The Tag Engine runs as a separate application, independent of the Human Machine Interface (HMI) application. Both the device servers and the HMI application communicate with the Tag Engine. Select **Tools»DSC Module»Launch Engine** to start the Tag Engine from your application. You also can click the button shown at left on the DSC Module Run-Time System window.

The Tag Engine performs the following tasks for the 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 to the Citadel database

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

Citadel Historical Database

The Citadel historical database is a National Instruments database used by the DSC Module and other National Instruments products that efficiently stores data acquired and processed by applications.

Refer to Chapter 5, *Historical Datalogging and Extraction*, for more information about the Citadel database.

Historical Data Viewer

Use the Historical Data Viewer to view the data stored in any Citadel database. It exists outside of the LabVIEW environment, in the MAX environment, and requires no programming. Select **Tools»DSC Module» View Historical Data** to open the Historical Data Viewer from your application. You also can click the button shown at left on the DSC Module Run-Time System window.

Refer to the *Viewing Historical Data* section of Chapter 5, *Historical Datalogging and Extraction*, for more information about the Historical Data Viewer.

The Classic Historical Trend Viewer (HTV) is still supported for legacy applications.

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 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»DSC Module»Security»Edit User Accounts** to open the User Account Manager from your application.

Refer to the *Creating and Editing User and Group Accounts* section of Chapter 6, *Security*, for more information about the User Account Manager.

Server Browser

In the DSC Module, a *device server* is an application that communicates with and manages input/output devices such as PLCs, remote input/output 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 2, *Servers*, for more information about device servers.

Use the Server Browser to browse 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»DSC Module»Advanced» Server Browser** to open the Server Browser from your application.

Customizing Your Work Environment

Complete the following steps to customize your work environment and to set startup options.

1. Select **Tools»DSC Module»Options** to display the **Options** dialog box from your application.
2. Select among the options on the **Environment**, **Startup**, and **Advanced** tabs.

   To view descriptions of these options, press <Ctrl-H> or select **Help»Show Context Help** and move the cursor over any field.
3. Click the **OK** button.

Refer to the **LabVIEW Environment»Customizing Your Work Environment** book of the *LabVIEW Help* for more information about customizing the LabVIEW work environment.

2

Servers

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

The DSC Module applications you run with the DSC Module Run-Time System might require one or more servers. Check with the application developer for information about any server software you might need to install and configure for your application to run.

The DSC Module can connect to any OPC-compliant server and to many third-party device servers. You also can connect to VI-based servers.

A server *item* is a channel, input/output point, or variable in a hardware device. Connect DSC Module applications to these server 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 Human Machine Interface (HMI) application writes that tag value and handle and report communications and device errors. A good device server covers all device- and hardware-specific details, establishing a device-independent input/output layer for the DSC Module. Many device servers include a configuration utility as well as the run-time application that communicates with the Tag Engine.

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

The DSC Module also can function as an OPC server and as a data source for the Logos networking protocol. Refer to the *LabVIEW Help* for information about the Logos networking protocol.

A server is not always the same as a device driver or an instrument driver. An instrument driver is a software component that is designed to control a programmable instrument such as a multimeter. A device driver is a

low-level software component that a computer needs to work with a plug-in interface. A device driver can also function as a server if it meets certain standards, such as the OPC specification.

Server Types

The 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 DSC Module* section later in this chapter for more information about DDE servers.
- **VI-based servers**—Use VIs to provide data to the Tag Engine. You also can use servers provided by a third-party hardware manufacturer.

Installing and Configuring Servers

After you select the device servers to use with the hardware, install and configure them according to the server documentation or your application 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 device servers manually before the DSC Module can 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 that ship with the DSC Module include a VI you use to register the server but should be already registered when you install the DSC Module.

Note VI-based servers are different from the LabVIEW VI Server. Refer to the *LabVIEW Help* for information about the LabVIEW VI Server.

Unregistering a Device Server

Unregister an OPC server by uninstalling the server software.

You can usually unregister VI-based servers in the Server Browser. Unregister the device server 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 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 the server with the DSC Module again.

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

1. Open the Server Browser by selecting **Tools»DSC Module» Advanced»Server Browser** from your application.
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 device server in your system, the DSC Module also registers the location of its configuration utility, if possible.

Note On Windows 2000/NT/XP, 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 by selecting **Tools»DSC Module» Advanced»Server Browser** from your application.
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. Refer to the *Viewing Information about Running Servers* section in this chapter for information about the Engine Manager.

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 by selecting **Tools»DSC Module» Advanced»Server Browser** from your application.
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. The **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.

OPC servers have an optional Server Browse Address Space Interface. If a server supports this interface, the 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. With the Tag Engine running, open the Engine Manager by double-clicking the Tag Engine icon in the Windows task tray.

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. The **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.

   OPC servers have an optional Server Browse Address Space Interface. If a server supports this interface, the 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.

Testing a Server

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

1. Open the Server Browser by selecting **Tools»DSC Module» Advanced»Server Browser** from your application.
2. Check the **Servers** listbox to see if the server is listed. If it is not, go to step 4.
3. Select the server to display and click **View Server Information** in 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 and save the tags, complete the following steps to make sure the server is providing data properly.

1. Start the Tag Engine by selecting **Tools»DSC Module» Launch Engine**.
2. Open the Tag Monitor by selecting **Tools»DSC Module» Monitor Tags**.
3. Double-click tags under the server in the left tree.
4. Make sure the data values and time stamps change.

If you cannot get live data with the Tag Monitor, use the configuration utility for that server to check the installation and configuration. Check the status of tag data in the **Quality** column of the Tag Monitor.

Connecting to Data Published by LabVIEW Real-Time Module Applications

Use National Instruments DataSocket technology to share live data with other VIs and other applications. DataSocket pulls together established communication protocols for measurement and automation in much the same way a Web browser pulls together different Internet technologies.

You can configure a host computer DataSocket server for use with the LabVIEW Real-Time (RT) Module. Refer to the *LabVIEW Help* for

information about how to configure a host computer DataSocket server for use with LabVIEW.

In this case, the RT Module publishes its real-time data to the DataSocket server on the host machine. You then can use the DSC Module to create tags that connect to the data in the DataSocket server running on the host computer.

Using Other Remote Servers

Using a DSC Module application, you can access data from any FieldPoint FP-16*xx* or FP-20*xx* module running on a computer in the network that is running a National Instruments networking protocol, such as Logos or DataSocket. Refer to the *LabVIEW Help* for information about Logos and DataSocket proprietary networking protocols.

To access LabVIEW applications as servers, you must register the computers on which they are running. Refer to the *Registering Network Computers* section of Chapter 7, *Networking and Running Applications*, for more information about registering computers.

Using DDE Servers with the DSC Module

The 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 connect to.

Third-party DDE servers do not register themselves with the DSC Module. Therefore, the 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 DSC Module returns system error messages if it cannot connect to the DDE server when it starts the Tag Engine. After returning system error messages, the DSC Module attempts to reconnect to the DDE server periodically.

3

Using Tags to Manage Input/Output 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 input/output point. You also can use a *memory tag* for data held by an application that you need to use or track. Refer to the *Determining When to Use Memory Tags* section later in this chapter for more information about using memory tags. A *network tag* is 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 tag data is scaled, if and how a tag is logged to a historical database, and alarm levels and priorities for tag data.

By configuring tag attributes, you can accomplish the following tasks.

- 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
- 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»DSC Module»Configure Tags** from your application. You also can click the button shown at left on the DSC Module Run-Time System window. Before you create or configure tags, you must install and configure the servers. Refer to Chapter 2, *Servers*, for more information about installing and configuring servers.

The application developer configures the tags for your application. However, you might have to adjust or maintain the tags, as described in this chapter.

Configuration Files

After you create tags and configure their attributes, you save that information in a configuration (`.scf`) file. Any 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 application and does not need to be specific to any single application. Multiple applications can run concurrently using the same `.scf` file.

Changing the Active .scf File

The active (default) `.scf` file is the last `.scf` file you saved with the Tag Configuration Editor, except for the first time you run the DSC Module. 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.

Changing the Active .scf File Manually

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

1. Select **Tools»DSC Module»Configure Tags** from your application.
2. Select **File»Open** to open a `.scf` file.
3. Select **File»Save** or **Save As** to save the `.scf` file.

Creating Tags

You can create tags in the following ways.

- Generate tags automatically in the Tag Configuration Wizard.
- Create tags manually in the Tag Configuration Editor.

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.

Note You might not be able to create new tags because you cannot modify your application. However, you might find it necessary to modify the attributes of a tag using the same tools you use to create a tag.

Generating Tags with the Tag Configuration Wizard

Use the Tag Configuration Wizard to generate tags from the server information if you want the Tag Engine to monitor a large number of input/output points in the system. When you run the server configuration utilities for the servers on the system, you can define devices and items for the input/output 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, input/output group, input/output connection, and scaling attributes for each server item to create the tags. For 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.

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

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Click the **Configuration Wizard** button in the toolbar, shown at left.

3. Expand each server branch in the **Servers** listbox to display the devices and items for one or more servers. If the **Servers** listbox lists item ranges instead of individual item names, go to step 7.
4. Select the items for which you want to create tags. Select a branch in the **Servers** listbox 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) Create a tag for a DataSocket item:
   a. Click the **DataSocket** button.
   b. Double-click **DataSocket Server** to see the items on the local computer, or browse to a network computer under **My Network Places** and view its DataSocket items.

Note To browse the DataSocket server, the server must be running on the local computer.

   c. Select the item you want to create a tag for. You can create only one tag at a time. The tag created from this item uses the entire

URL as its tag name (the "/" is replaced with "_"), with the data type and access rights shown.

d. Click the **OK** button.

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 the 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 in the **Create This Many Items** edit box.

      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 input/output 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 in the **Select Items for Automatic Tag Generation** dialog box.
   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.
   e. Click the **OK** button.

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. Complete the following steps if you want to change the tag name format for a server.
    a. Select a server in the **Select Items for Automatic Tag Generation** dialog box.
    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.
    e. Click the **OK** button.
11. To remove individual items from the **Selected Items** listbox, select the items in the **Selected Items** listbox and click the **Remove Item(s)** button. To remove all items, click the **Remove All** button.
12. When all 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 new and 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 manually change the configuration of any tag later. Complete the following steps to create tags manually.

1. If you have not already, install and configure the server as described in Chapter 2, *Servers*.
2. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 `.scf` file. Refer to the *Configuring Tag Attributes* section for 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

Complete the following steps to import tags into a local `.scf` file from a `.scf` file located on another computer.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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** path control and navigate to a `.scf` file on any computer on the network.
4. Click the **Add** or **Add All** button, or select the tags individually and click the **Add** button to add the tags import to the **Selected Tags** listbox.
5. Click the **Import** 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 DSC Module tags into an existing VI-based DAQ application. Using this method, you can take advantage of the DSC Module features such as alarming, logging, and security, without reconfiguring the virtual channels. Complete the following steps to import DAQ virtual channels as tags.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **File»Import DAQ Memory Tags**.
3. Click the **Add All** button or select the DAQ channels you want to import and click the **Add** button.
4. Click the **OK** button. The DSC Module creates memory tags with the same names as the DAQ channels.

Editing Tags

You can edit tags manually or in a spreadsheet.

Editing Tag Configuration Manually

When you create a tag using the Tag Configuration Wizard, the wizard assigns the default values for each tag attribute. 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 by selecting **Tools»DSC Module»Configure Tags** from your application.
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 any static attributes have been changed, the Tag Engine shuts down and restarts to update static attribute values. If you have changed only dynamic attributes in the `.scf` file, the Tag Engine updates without restarting.
6. Use the Tag Monitor to test the tag configuration and make sure you are reading and writing data properly with the servers. Select **Tools»DSC Module»Monitor Tags** to launch the Tag Monitor.

Caution Communication between the Tag Engine and any device server stops temporarily when the Tag Engine shuts down and restarts.

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 that you can open with any text editor program, such as Notepad or Microsoft Word.

If you use spreadsheet files with the Tag Configuration Editor, consider the following guidelines:

- 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 the data.
- 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 or tag-delimited text 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

Complete the following steps to export tag configuration to a spreadsheet.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **File»Export** to launch the **Select Tag Fields for Export** dialog box.
3. In the **Spreadsheet File** edit box, enter the full path to the tab-delimited text file (`.txt`) you want your tags exported to.
4. 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.
5. Click the **OK** button.
6. In a spreadsheet application, open the text file you created.

Importing Tag Configuration from a Spreadsheet

Complete the following steps to import tag configuration from a spreadsheet.

1. Save the spreadsheet as a `.txt` file.
2. In LabVIEW, open the Tag Configuration Editor by selecting **Tools»DSC Module»Configure Tags** from your application.
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 then are 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.

Complete the following steps to define default tag configuration values.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module»Configure Tags** from your application.
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 you create a new tag, import a tag from a server, or import 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. Only if the difference between the new value and the previous value exceeds the deadband does the new value replace the old.

Note Deadbands in 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. By increasing the deadband size, you can reduce the strain on the Tag Engine, though this might compromise data resolution.

Beware that if you set the update deadband too high, the Tag Engine might not be updated, resulting 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

You can control three deadband settings when you configure individual tags: update deadband, log deadband, and alarm deadband.

The update deadband affects how the Tag Engine updates values. Log deadband and alarm deadband values both operate on the values that have passed through the update deadband value. If the update deadband value is set too high, it can interfere with the alarm deadband and log deadband settings.

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 and affects values coming from the server before the Tag Engine gets the value. So the effects of this deadband setting can ripple through the update deadband, log deadband, and alarm deadband values. 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 the 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. The new value replaces the existing value only when the difference between the new value and the existing value exceeds the update deadband. Set the update deadband value in the **Operations** tab of the **Tag Configuration** dialog box (in the Tag Configuration Editor, select **Edit»Edit Tags**).

For 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 Tag Engine is 12.3. If the Tag Engine reports a new value of 13, the Tag Engine does not update because the change in value did not exceed the low deadband. If the Tag Engine reports a new value of 11, it updates because the difference is greater than the low deadband.

Setting Log Deadbands

When you set a log deadband value, the new value is compared to the old value. The new value is logged if it exceeds the log deadband value. Set the log deadband value in the **Operations** tab of the Tag Configuration Editor.

The default setting for log deadband is 1%.

For example, for a data point with a range of values of 0 to 100, set the log deadband value to 2%. The last value logged was 12.3. When the Tag Engine updates to 11, the updated value is not logged because it is smaller than the 2% deadband. The value in the Tag Engine 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 value, the new value is compared to the old value. The alarm is triggered when the value falls outside the range of the deadband and is cleared when the alarm value reaches the inside range of the deadband. Set the alarm deadband in the **Alarms** tab of the Tag Configuration Editor.

For 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%. The alarm condition is not triggered until the Tag Engine value drops to 12 or below. The alarm stays active until the Tag Engine 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 update deadband again before logging the values.

Complete the following steps to set I/O group deadbands for an OPC server.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 the deadband setting. Refer to the OPC server documentation for more information about that server.

Deleting Tags

Complete the following steps to delete tags.

Caution Deleting tags might cause the application to stop working. Consult the application developer before deleting tags.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select the tag you want to delete.
3. Click the **Delete Tag** button, shown at left, on the toolbar or press the <Delete> key. 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.

Click the **Delete Tag** button in the toolbar 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 Configuration Wizard, the Tag Configuration 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 by selecting **Tools»DSC Module» Configure Tags** from your application.
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 input/output point. Select **Memory** from the **Tag Access** pull-down menu. 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. A diamond appears next to the tag to indicate it has changed.
5. Select **File»Save** to save the changes.

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

If any static attributes have been changed, the Tag Engine shuts down and restarts to update static attribute values. If you have changed only dynamic attributes in the `.scf` file, the Tag Engine updates without restarting.

6. Use the Tag Monitor to test the tag configuration and make sure you are reading and writing data properly with the 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 depending on the data type. The tag data types are analog, discrete, bit array, and string tags.

Analog Tags

An analog tag is a continuous value representation of a connection to a real-world input/output 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, such as 0 to 100.

Discrete Tags

A discrete tag, such as a Boolean control or indicator in LabVIEW, is a two-state (ON/OFF) value representation of a connection to a real-world input/output point or memory variable. This 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 input/output point or memory variable. This type of tag can be composed 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 input/output 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 Tag 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 input/output 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.

Refer to the *LabVIEW Datalogging and Supervisory Control Module Run-Time System Help* for more information about static and dynamic tag 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 4, *Alarms and Events*, for information about alarm groups. Complete the following steps to define a tag group.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Double-click a tag to display the **Tag Configuration** dialog box.
3. Click the **General** tab.

4. Select **Enter New** from the **Tag Group** pull-down menu to select an existing tag group or define a new tag group.
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

Use I/O groups to configure 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 you are creating a tag, you must create an I/O group before connecting the tag to a server item.

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

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 **Device** pull-down menu.

     A list of items connected to that device appears in the **Item** pull-down menu 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 the server before the DSC Module can access the server. Refer to the *Installing and Configuring Servers* section of Chapter 2, *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

Complete the following steps to configure DDE devices and items.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 the **Add** button is not 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 **Enter Device Name** textbox. For example, type `excel|worksheetname` to connect to a a specific 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 the 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 the DDE server documentation for more information about item names.

10. Click the **OK** button.
11. 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 by selecting **Tools»DSC Module» Configure Tags** from your application.
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 by selecting **Tools»DSC Module» Configure Tags** from your application.
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, 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 by selecting **Tools»DSC Module» Configure Tags** from your application.
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 in the **Item Connection** section of the **Connection** tab. If the server does not support item names, these buttons are disabled.
   - To add an item name, click the **Create** 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 OPC servers. Complete the following steps to configure item resources.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 in the **Item Connection** section of the **Connection** tab. 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** or **Create** button. In the configuration dialog box, configure a new item for a selected server and click the **OK** button. The **Add** button is available for servers where new items can be added. For servers with a fixed number of items, the **Create** button is available.
   - 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, specific to the server, 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.

5. 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.
6. Click the **OK** button.
7. Select **File»Save** to save the changes.

Configuring a Tag to Log Data or Events

By default the DSC Module enables logging when you create a tag. When you start logging, you log all tags except those you have configured not to be logged. To start logging, either manually activate logging in the Engine Manager or set logging to begin automatically when the Tag Engine starts.

Complete the following steps to configure logging manually.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Double-click the tag to display the **Tag Configuration** dialog box.
3. Click the **Operations** tab.
4. Place or remove a checkmark in the **Log/Print Events** and **Log Data** checkboxes. Events in this case include enabled alarms for the tag. If **Log Deadband** is available, set the logging deadband and the data resolution.
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 the *Logging Historical Data* section in Chapter 5, *Historical Datalogging and Extraction*.

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

Initializing Tag Values

Complete the following steps to initialize a tag to a known value when the Tag Engine starts.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Double-click a tag to display the **Tag Configuration** dialog box.
3. Click the **Operations** tab.
4. Place a checkmark in 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 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 application.

Complete the following steps to scale analog tags.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 *(mx + b)* between raw and engineering ranges. Select **Square Root** to enable a square root conversion between the raw (RawMin and RawMax) and engineering (EngMin and EngMax) 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 to the square root of the raw unit after compensating for offsets.

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 Table 3-1.

Table 3-1. Square Root and Linear Scaling Example Values

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

Table 3-1. Square Root and Linear Scaling Example Values (Continued)

Raw Units	Linear Scale	Square Root Scale
90	9	9.49
100	10	10

Table 3-2 shows a raw value that ranges from 0 to 100 scaled to engineering units ranging from 15 to 30. Offsets deliver more complicated results.

Table 3-2. Scaling with Offset Example Values

Raw Units	Linear Scale	Square Root Scale
0	15	15
4	15.60	18
10	16.50	19.74
16	17.40	21
20	18	21.71
30	19.50	23.22
36	20.40	24
40	21	24.49
50	22.50	25.61
60	24	26.62
64	24.60	27
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 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 V 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 to 20 mA readings. The actual flow is measured in gallons per minutes (GPM). 4 mA corresponds to 0 GPM and 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 Units** 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. Complete the following steps to scale discrete tags.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from an open VI window with the **Tools** menu visible.
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. Complete the following steps to scale bit array tags.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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, reads the bit array

   `10110011100011001100111000110010`

   as the following:

   `00000000000000001100111000110010`
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 3-3 shows examples of tags configured for bit array scaling.

Table 3-3. 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

Use alarms to notify 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 DSC Module, 15 is the highest priority and 1 is the lowest.

Alarms are triggered by two main factors: status and 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 4, *Alarms and Events*, for information about accessing and displaying alarms and events.

Complete the following steps to set alarms.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 data 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 a significant change. For example, if a tag that represents a temperature value hovers near an alarm limit of 40 °C, the tag might go in and out of alarm many times in a relatively short period of time. Table 3-4 shows examples of events with **Alarm Deadband** set to 0.0%.

Table 3-4. 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 frustrate operators who have to acknowledge alarms even when change is insignificant. Increase the alarm deadband to alleviate this problem.

For the tag to go into alarm, it must exceed the exact alarm value (in the previous example, 40). However, to be considered normal again, it must deviate from the alarm value by an amount greater than the alarm deadband. For example, if the range is 0 to 100 °C, an alarm deadband of 1.0% (one degree Celsius) eliminates unnecessary events. Table 3-5 shows examples of events with **Alarm Deadband** set to 1.0%.

Table 3-5. 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 **Alarm Invert Mask** to determine which bits generate an alarm on low (OFF) rather than generating an alarm on the default value, high (ON). You can use **Alarm Select Mask** (logical AND) to determine which bits to consider 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 support only 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**. The default is **Auto Ack on Normal**.

- **Auto Ack on Normal**—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.
- **User Must Ack**—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 software-generated values or combinations of values from different input/output tag readings. You do not need to use memory tags for program variables unless you want to use the historical and event logging or alarm management capabilities of the Tag Engine.

Creating a Memory Tag

Complete the following steps to create a memory tag.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **Edit»Create** and select the type of tag you want to create.
3. Click the **Connection** tab.
4. Select **Memory** from the **Tag Access** pull-down menu.
5. Select any other settings you want for the memory tag.
6. Click the **OK** button.
7. Select **File»Save** to save the changes.

Note You might not be able to create new tags because you cannot modify your application. However, you might find it necessary to modify the properties of a tag.

Customizing the Tag Configuration Editor View

Complete the following steps to customize the Tag Configuration Editor view.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **Edit»Column Setup** to display the **Select Tag Fields to View** dialog box. You can display the same columns of information that describe 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 columns. You also can drag and drop fields to rearrange them.
5. Click the **OK** button to close the **Select Tag Fields to View** dialog box and return to the Tag Configuration Editor. The tag fields you selected to view appear in the Tag Configuration Editor tag list.

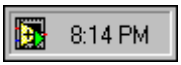
Accessing Tags Over a Network

A 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 DSC Module servers. A DSC Module server also can act as a client and get its data from other DSC Module server computers.

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

Access to data through DataSocket or across the network is subject to security access rights. Refer to the *Setting Data Access* section in Chapter 6, *Security*, for more information.

Viewing Tag Engine Status



The Engine Manager shows the current state of the Tag Engine. Select **Tools»DSC Module»Launch Engine** to launch the Tag Engine from your application 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 the Windows taskbar, shown at left. Double-click the Tag Engine icon to open the **Engine Manager** dialog box.

You can leave the **Engine Manager** dialog box 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.

Refer to the *LabVIEW Datalogging and Supervisory Control Module Run-Time System Help* for information about Engine Manager settings.

Configuring Tag Engine Parameters

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

Note Although you can configure these parameters, National Instruments highly recommends that 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.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **Configure»Engine**. The **Engine Configuration** dialog box appears.
3. Select among the Tag Engine options on the various tabs.
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

Use the Tag Monitor to monitor the value, time stamp, alarm state, and quality for selected tags in the system, as well as write the value to an output or input/output tag. Complete the following steps to monitor and write tag values.

1. Open the Tag Monitor by selecting **Tools»DSC Module» Monitor Tags** from your application. The **Tag Monitor** dialog box appears.
2. Place a checkmark in the **Start Tag Engine when Tag Monitor starts** checkbox to configure the DSC Module to start the Tag Engine each time you launch the Tag Monitor.
3. Click the **OK** button.
4. 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 in the local computer, under **My Computer**, and across the network, under the computer name and **My Network Places**. You also can see data from other software and devices on network computers. To find tags or data on another computer, you must

first register the computer. Refer to the *Registering Network Computers* section of Chapter 7, *Networking and Running Applications*, for more information about registering and unregistering computers.

5. 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.
6. 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.
7. Select **View»Refresh** to refresh the listbox in the left pane and the alarm view in the bottom right pane. The tags in the tag display pane update continuously, so you do not need to refresh the view.
8. Select **File»Save As** to save different sets of tags to monitor.

4

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 DSC Module application. Events can be divided into two groups: *tag events* that pertain to individual tags, and *system events* that pertain to the overall 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 DSC Module, an *alarm* is a specific kind of event related to the value of a tag. An event can 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 3, *Using Tags to Manage Input/Output in LabVIEW*, for more information about setting alarm attributes. You also enable tag event logging when you configure tags. Refer to the *Configuring a Tag to Log Data or Events* section of Chapter 3, *Using Tags to Manage Input/Output in LabVIEW*, for more information about enabling tag event logging.

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

The application developer configures the alarms and events for your application. However, you might have to adjust or maintain the alarms and events, as described in this chapter.

Logging and Printing for Alarms and Events

Complete the following steps to configure automatic logging and printing for alarms and events in the Tag Configuration Editor.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **Configure»Events** to display the **Event Configuration** dialog box.
3. Click the tabs to select among the logging and printing options for alarms and events. To open the **Context Help** window and view descriptions of these options, press the <Ctrl-H> keys and move the cursor over any field.

Note If you log data to directories created in a secure file system, such as NTFS, 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, the DSC Module is 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.

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 DSC Module applications. Some approaches operate through traditional VIs and others use capabilities built into National Instruments networking. The differing approaches are provided for flexibility and compatibility with existing applications.

- You can use the Tag Monitor to view alarms and events. Refer to the *Monitoring and Writing Tag Values* section of Chapter 3, *Using Tags to Manage Input/Output 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 Tag Engine and are then stored in the Citadel database.

- View alarms and events logged to a Citadel database with the Historical Data Viewer in the Measurement & Automation Explorer (MAX) environment. For more information about Historical Data Viewer, refer to the *Historical Data Viewer Help* in MAX.

Viewing Alarms and Events with the Alarm & Event Display Control

The Alarm & Event Display control is the easiest way to monitor and acknowledge alarms and events. The alarm & event display appears in the Tag Monitor and might also be on the front panel of your HMI application. The Alarm & Event Display control shows alarms and events from every computer and process you configure it to display.

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. Complete the following steps to select the source of the alarms to monitor.

1. While in run mode, 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

Complete the following steps to set the filter criteria to see certain alarms and events in the Alarm & Event Display control.

1. In run mode (**Operate»Change to Run Mode**), right-click the Alarm & Event Display control on the front panel and select **Filter Options**.

2. Select among the filter options.
   - Place a checkmark in the **Priority** checkbox and type values in **Min** and **Max** to monitor alarms with specific priorities.
   - Place a checkmark in the **User Name** checkbox and type a user name 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.
   - Place a checkmark in the **Ack User Name** checkbox and type a user name 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.
   - Place a checkmark in the **Ack Comment** checkbox and type a comment to restrict the alarms displayed to those with that acknowledgement comment.
   - Place a checkmark in the **Object Name** checkbox and type a tag name to restrict alarm monitoring to alarms involving that tag name. You can enter only one tag name at a time, but you can use wildcards to widen the scope of the alarms reported. You must enter a completely qualified tag name, as displayed in the tag display pane above the alarm view.
   - Place a checkmark in the **Description** checkbox and type a description to restrict monitoring to alarms that meet a criteria. You can select only one description category at a time, but you can use wildcards to widen the scope of the alarms reported.
   - Place a checkmark in the **Area Name** checkbox and type an area name to restrict monitoring to that alarm area. You can enter only one alarm area at a time.
   - Use the fields in the **Old Alarms** section to display alarms after they have been acknowledged.
   - Select a **Show** option to display alarms only, events only, or both alarms and events.
   - 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 the 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 Listbox

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.

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.

The front panel of your HMI application should contain an **ACK** button for you to use with the alarm summary display listbox.

Using an Event History Display Listbox

An *event history* is a collection of all the alarms and events pertaining to tag values that have occurred in the DSC Module since the Tag Engine started.

The front panel of your HMI application should contain an **ACK** button for you to use with the event history summary display listbox.

Viewing System Errors and Events

System errors are conditions on a system level (as opposed to the tag-level) that result in problems with the DSC Module functioning. When a system error occurs, LabVIEW prompts you with a dialog box. You can turn this dialog box on or off from the Engine Manager.

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 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. The DSC Module creates a new `.log` file each time the Tag Engine is launched.

5

Historical Datalogging and Extraction

Data processed by the Tag Engine is contained in memory, and there is no file created to hold that data. When the Tag Engine is stopped, it retains the last data received, but it does not update any values until it begins to run again. Because data logged to the Citadel historical database is logged by the Tag Engine, no data will be logged to the Citadel database 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 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 a Citadel Database*, for more information about using the Citadel ODBC driver.

The maximum size for alarm and event storage is 2 GB. Once the 2 GB limit is reached, you must log to a new database. You can have up to 16 active (attached) databases on one computer. Under Windows 2000/NT/XP/98, the Citadel service runs on the computer as a service.

Caution Do *not* stop this service while the DSC Module Run-Time System or the Tag Engine is running.

Data you configure to be logged to a Citadel database 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 the local computer or to a remote computer on the network, but the directory to which you want to log must be writable from the computer running the Tag Engine.

Access Citadel data through the Historical Data Viewer, 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 tags for logging as described in the *Configuring a Tag to Log Data or Events* section of Chapter 3, *Using Tags to Manage Input/Output in LabVIEW*.
2. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
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 the <Ctrl-H> keys and move the cursor over any field.
   - To set automatic data logging when the Tag Engine starts, select **Configure»Historical** and place a checkmark in the **Start logging on system start-up** checkbox. To set event logging when the Tag Engine starts, select **Configure»Events** and place a checkmark in the **Start logging on system start-up** checkbox.

     When the Tag Engine loads a `.scf` file with these settings configured to start historical logging, the Tag Engine logs data and events from the moment it starts, and checkmarks are placed in the logging option checkboxes in the Engine Manager. This setting is part of the `.scf` file and can change when you change active `.scf` files.
   - If you have not configured the `.scf` file to initiate logging when the Tag Engine starts, you can start logging manually through the Engine Manager by placing checkmarks in the appropriate checkboxes.

   In both cases, the DSC Module logs data from all tags that have been configured for logging.

Note If you log data to directories created in a secure file system, such as NTFS, grant the System account Change or Full Control permissions to the directory. If you do not grant the System account appropriate access to the directory where the database to log from resides, the DSC Module is 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 by selecting **Tools»DSC Module» Launch Engine** while the Tag Engine is running.
8. Place checkmarks in the **Log** checkboxes to turn on historical logging.

Logging Data in Sets

The DSC Module allows you to accomplish *batch logging* by logging and retrieving data sets. A data set is a group of tag values that are logged together as a set during some finite time period.

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.

To log and retrieve data in sets, configure the DSC Module to track the data sets, then use the Historical Data Viewer in MAX to retrieve the data set values.

Note You cannot include a data set within another data set.

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 you access completed data sets.

Creating a Data Set for Logging

Complete the following steps to create a data set for logging.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application and make sure the active `.scf` file contains the tags you want to log in the data set.
2. Save the `.scf` file if you have added or removed tags since the last time you saved.
3. Select **Servers»Data Set Logger Configuration**. 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, so the DSC Module creates 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 time stamp.

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

   - (Optional) **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 if the previous run has ended. When the end condition is met, the run ends. Use the values in Table 5-1 to programmatically change the start and stop conditions.

Table 5-1. 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 value changes from FALSE to TRUE, a new run starts.
2	**Discrete Tag OFF**—When the specified discrete tag value changes from TRUE to FALSE, a new run starts.
3	**Analog Tag > Limit**—When the specified analog tag value exceeds the user-provided limit, a new run starts.

Table 5-1. Data Set Run Start/End Conditions (Continued)

Value	Start/End Condition
4	**Analog Tag = Limit**—When the specified analog tag 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 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 value equals the user-provided string, a run starts. Use only text strings for this option.
7	**Time of Day**—When the system 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 the **Add** button to select the tags you want to include in the 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 the **Remove** button.

Note Use a data set configuration only with the `.scf` file you used to create the data set. If you change the name of a tag in a `.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 do not show up in the data set configuration tool until the `.scf` file is saved.

6. (Optional) Click the **Advanced** tab, and click the **Add** button 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 the **OK** button.
8. Create at least one tag connection from the `.scf` file to a Data Set Logger server item to ensure that the Data Set Logger server will run, because it will be launched by the Tag Engine.

   a. To connect 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.

b. Select the items from which you want to create tags under the data set on the left, and click the **Add Items** button. Create at least one tag from the server items. You might find the **active**, **currentid**, and **internal_id** server items are most useful. Refer to the *LabVIEW Datalogging and Supervisory Control Module Run-Time System Help* for a description of each data set logger server item.

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

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

Editing Data Sets for Logging

Complete the following steps to edit an existing data set.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application and make sure the active `.scf` file contains the tags you want to include in the 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** and navigate to the data set you want to edit. Click the **OK** button.
3. Make changes to the configuration of the data set. Refer to the *Creating a Data Set for Logging* section earlier in this chapter for more information about the configuration options.
4. Restart the Tag Engine if it is running so that the changes take effect.

Considerations for the Data Set Logger

Nested data sets are not allowed. 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 you access completed data sets.

Retrieving Logged Data Sets

You can use the Historical Data Viewer in MAX to retrieve data that has been logged in sets. Refer to the *Historical Data Viewer Help* in MAX for more information about retrieving historical data.

Archiving Historical Data

You can archive historical data in the following ways.

- Use the archiving feature of the Historical Data Viewer in MAX.
- Archive the database manually by copying or moving the files. You must detach the database before you archive it manually. Detach the database using the Historical Data Viewer in MAX or the Detach Database VI. Then use the Historical Data Viewer to reattach the database after you copy it.

  When you decide to archive historical data manually, copy the `.scf` file and 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. 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. 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`.

  Copy or move all files in the database directory to the directory you have selected as the logging directory. However, if you copy those files into a directory with an already existing database, the file names will collide.

Converting a .scf File Created in an Earlier Version of the DSC Module

To convert a `.scf` file created in an earlier version of the DSC Module, open the `.scf` file in the Tag Configuration Editor. The **SCF File Version Mismatch** dialog box appears to direct you through the conversion process.

To convert previous versions of Citadel database files to the current version of the Citadel historical database, use the Historical Data Viewer in MAX.

Databases Associated with a .scf File

When you open a `.scf` file created in a previous version of the DSC Module, a conversion wizard launches. After converting the `.scf` file, the wizard gives you the option of converting existing Citadel 3.*x* or 4 databases that are associated with that `.scf` file. We recommend using the default conversion options. If you must save the converted database to a location other than the default path due to a shortage of hard drive space, for example, you might cause problems in the application.

If you attempt to log data from a converted application without first converting the database to Citadel 5, the DSC Module creates a new, empty Citadel 5 database in the same location as the Citadel 4 database and logs data to that new database. To get all the logged data back into one database, you must use Historical Data Viewer to convert the Citadel 4 database and then merge it with the new Citadel 5 database.

Databases Not Associated with a .scf File

If you choose not to convert a Citadel database when the wizard prompts you or if you have an existing database that is not associated with a converted `.scf` file, you can convert the database using the Historical Data Viewer in MAX. In LabVIEW, select **Tools»DSC Module»View Historical Data»Historical Data Viewer**. In MAX, right-click the **Historical Data** category, select **Create New**, then select the appropriate conversion in the dialog that appears.

Remote Databases

You cannot convert a database remotely or over the network. You must convert it on the computer where it exists.

Viewing Historical Data

There are two methods for viewing historical data that has been logged to a Citadel historical database.

- Use the Historical Data Viewer. The Historical Data Viewer exists in MAX and requires no programming. The Historical Data Viewer allows you to save multiple views of traces and settings.

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

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

Your application might offer an alternative way to view historical data. Refer to the application documentation or consult the application developer for more information.

Printing Historical Data

You can print logged historical data in the following ways.

- Print historical data trends from the Historical Data Viewer in MAX, which allows you to export data to a spreadsheet or to HTML format.
- 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 a Citadel Database*, for more information about using ODBC applications with the Citadel database.

Your application might offer an alternative way to print historical data. Refer to the application documentation or consult the application developer for more information.

6

Security

Your application implements security with user and group accounts.

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.

For example, the application developer controls access to a front panel control by giving access to individual user or group accounts.

Creating and Editing User and Group Accounts

You might need to create, delete, or edit user accounts. 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 DSC Module 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 the network, you must use the same `lookout.sec` file for all installed copies of the LabVIEW Datalogging and Supervisory Control (DSC) Module. Refer to the *Duplicating Security Files for Network Computers* section of Chapter 7, Networking and Running Applications, for more information.

Creating User Accounts

1. Open the User Account Manager by selecting **Tools»DSC Module» Security»Edit User Accounts** from your application.
2. Select **User»New User Account**.
3. Enter the domain name of the new user in the **Username** textbox.
4. Enter the **Full Name** of the user.
5. Enter job titles or other relevant information in the **Description** textbox.
6. Enter the user password in the **Password** textbox.

7. Enter the password a second time in the **Confirm Password** textbox 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. Select **Password Expires** and enter a value to 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.
10. Place a checkmark in the **Account Disabled** checkbox if you want to disable a user account without removing the user from the system.
11. 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.
12. To enter a user in a group, highlight the group in the **Not Member of** list and click the **Add** button. To remove a user from membership in a group, highlight a group in the **Member of** list 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.

13. Click the **OK** button.

Creating Groups

1. Open the User Account Manager by selecting **Tools»DSC Module» Security»Edit User Accounts** from your application.
2. Select **User»New Local Group**.
3. Assign a name to the group in the **Group Name** textbox.
4. Enter a description of the group in the **Description** textbox.
5. Assign the security level for members of this group in the **Security Level** pull-down list.

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 the local domain.
8. Highlight the names you want to add in the **Names** field, and click the **Add** button to add those users to the group.

Modifying User and Group Accounts

The dialog boxes for editing users and groups are similar to the dialog boxes for creating users and groups. Complete the following steps to modify user and group accounts.

1. Open the User Account Manager by selecting **Tools»DSC Module» Security»Edit User Accounts** from your application.
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 do the **New User Account** 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 built-in user accounts and groups. 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. LabVIEW defaults to the Nobody account when no authorized user is logged on. The Nobody account always has a security level of 0.

You can edit all the properties of the Guest user account and the properties of the Guests, Operators, and System Operators groups.

Logging In and Out

To log in, select **Tools»User Name** from your application. You also can click the button shown at left on the DSC Module Run-Time System window. Type your user 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»DSC Module»Security»Logout**, or select **Tools»User Name** and click the **Logout** button.

Accessing User Information

After you log into the DSC Module, you can find out what your user privileges are, along with other user information, by selecting **Tools» DSC Module»Security»User Info** from your application.

The **Account** 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. Complete the following steps to change your password if you are not an Administrator.

1. Log in (**Tools»User Name** from your application).
2. Select **Tools»DSC Module»Security»Change Password**.
3. Type your old password, your new password twice, then click the **OK** button.

Restricting Access to the LabVIEW Environment

After you set up user and group accounts, you can implement security in several ways. You can configure access to most DSC Module utilities and the Tag Engine on a per-user or group basis. Security set up by selecting **Tools»DSC Module»Options** or **Tools»DSC Module»Security** applies to everything in the LabVIEW environment. Security set up in the Tag Configuration Editor applies only to the `.scf` file.

Setting Permissions for Accessing Tools

Complete the following steps to set permissions for the Tag Configuration Editor, Tag Engine, Tag Monitor, Server Browser, startup VIs, or server tester.

1. Log in as an administrator (**Tools»User Name** from your application).
2. Select **Tools»DSC Module»Options**.
3. Click the **Tools Access** button on the **Advanced** tab. The **Tools Access** dialog box appears.
4. Click the tab for the utility for which you want to configure permissions and click the **Edit** button. The standard 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.
5. Highlight the user or group you want to add and click the **Add** button. Set the **Access Rights**, and click the **OK** button when you finish.

Configuring Access to a Specific Tag

Complete the following steps to configure access to a specific tag.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Double-click the tag to display the **Tag Configuration** dialog box.
3. On the **General** tab, click the **Security** button to display the **Tag Data Access Rights** dialog box.
4. Set the **Access rights**. Click the **Help** button in the **Access Rights** dialog box for more information about the access rights fields.
5. Click the **OK** button twice.
6. Select **File»Save As** to save the changes.

If you do not use a specific setting for a tag, the tag inherits the data access settings made for the `.scf` file. Refer to the *Setting Data Access* section later in this chapter for more information about data access settings.

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. Complete the following steps to set `.scf` file access.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.

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 listbox 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 the user or group and click the **Remove** button.
   - To change a user or group permission, select the permission 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 the **Help** button in the **Access Rights** dialog box for more information.
5. Click the **OK** button when you are finished.

Setting Data Access

You can specify which users, groups, or computers are allowed to access a `.scf` file tag data from DataSocket, Tag Monitor, or other Tag Engines. You also can 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 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 account.
   b. If the user is not recognized, the proxy user settings are used.

The following examples demonstrate how the 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 block diagram,

the Nobody account is used, so the proxy user access rights are invoked.

- A DSC Module user attempts access to tag data from across the network, and both machines have the same `lookout.sec` file installed—In this case, the DSC Module application applies the access rights assigned to that user account.
- A DSC Module user attempts access to 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 the DSC Module application, and the proxy user access rights are applied.
- Someone using a program other than the DSC Module attempts to access tag data from across the network—In this case, the user is unrecognized by the DSC Module application, and the proxy user access rights are applied. This also applies to a LabVIEW user without the DSC Module installed.
- A remote Tag Engine connects to tags in the local Tag Engine—In this case, the DSC Module application applies the access rights assigned to the *engine user* account as defined in the remote 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 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 `.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 network computer have the same `lookout.sec` file installed.

Complete the following steps to set network access for users, groups, and computers.

1. Open the Tag Configuration Editor by selecting **Tools»DSC Module» Configure Tags** from your application.
2. Select **Configure»Security**. The **Access Rights Configuration** dialog box appears.
3. On the **Data Access** tab, click the **Edit** button. The **Host and Tag Data Access Rights** dialog box appears.

4. Configure the network security for user or group access. The listbox 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 the user or group and click the **Remove** button.
   - To change a user or group permission, select the permission and select from the options in the **Access** listbox.
5. To add a new user or group, click the **Add** button. The **Add Users and Groups** dialog box appears. Select the user or group you want to add and click the **Add** button. Set the **Access rights**. Click the **Help** button in the **Access Rights** dialog box for more information. Click the **OK** button when you are finished.
6. Configure the network security for *host access*. Host access controls whether a particular computer can access data on your computer, no matter who is logged on that computer. 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 network computers.
   - You can use the asterisk wildcard character to enable or disable all computers or all computers in a set. For instance, enter `*.yourdomain.com` to select all the computers with that domain address. This is the same syntax used by the VI Server TCP/IP Access page, available by selecting **Tools»Options**, then selecting **VI Server:TCP/IP Access**. Refer to the *LabVIEW User Manual* 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 **Select Computer** dialog box appears. Double-click the network name 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.
7. Click the **OK** button twice, then the **Close** button.

Setting a Proxy User Account

The proxy user is the account used for unrecognized access to tag data. For example, if you set the proxy user as Guest (default), then any unidentified client who attempts to access data is given the access rights you have assigned to the Guest account. Complete the following steps to set proxy user access.

1. Select **Tools»DSC Module»Options** and click the **Advanced** tab.
2. Click the **Proxy User** button to open the **Set Proxy User** dialog box.
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 the **OK** button.

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 the local Tag Engine has access to network tag data no matter who is logged in to the local 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 computer recognizes that account (if it is defined in its `lookout.sec` file), it grants the access rights defined for that account. If the remote computer 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 computer.

Complete the following steps to set up an engine user account.

1. Select **Tools»DSC Module»Options** from your application and click the **Advanced** tab.
2. Click the **Engine User** button to open the **Set Engine User** dialog box.
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 the **OK** button.

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

Complete the following steps to set access to the Tag Configuration Editor.

1. While logged in as Administrator, select **Tools»DSC Module» Options** from your application.
2. Click the **Advanced** tab, then click the **Tools Access** button.
3. On the **Tag Configuration Editor** tab, click the **Edit** button. The **Access Rights** dialog box appears.
4. The listbox 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 the user or group and click the **Remove** button.
   - To change a user or group permission, select the permission 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 the **Help** button in the **Access Rights** dialog box for more information.
5. Click the **OK** button when you are finished.

Setting Startup Login Options

You can set several login options, such as automatically logging in the last user when LabVIEW starts or setting a dialog box to appear at startup so you must log in manually. Complete the following steps to set startup login options.

1. Log in as an administrator (**Tools»User Name** from your application).
2. Select **Tools»DSC Module»Options**.
3. Click the **Advanced** tab.
4. Click the **Security Preferences** button to display the **Security System Settings** dialog box.
5. Click the **Startup** tab.
6. Set the login option to use when LabVIEW starts.
7. Click the **OK** button to close the **Security System Settings** dialog box and click the **OK** button to close the **Options** dialog box.

Disabling Special Keys

Complete the following steps to prevent anyone logged in below a certain security level from using 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»DSC Module»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 the **OK** button.

Note To disable special key access, you must have performed a **Custom** installation to install the **Keyboard Driver**.

7

Networking and Running Applications

This chapter describes how to set up applications for use on a network. You can run LabVIEW Datalogging and Supervisory Control (DSC) Module applications using the DSC Module Run-Time System, which has built-in support for the special DSC Module capabilities. You cannot run DSC Module applications using just the LabVIEW Run-Time Engine.

Setting up Network Applications

Complete the following steps to use a DSC Module application on a network.

1. Register the computers or devices on the network that use Logos networking.
2. Synchronize the clocks on all computers and devices.
3. Make sure the security files are compatible on the network computers.
4. Verify network paths and names.
5. Make sure you have the Citadel, Classified Ads, and Time Synchronization services running. Refer to the *Monitoring Services* section later in this chapter for more information about these necessary services.

Logos Networking Technology

National Instruments uses a proprietary networking technology known as the Logos networking protocol, installed as a service on the computer when you install the 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 the 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 it cannot act as a Logos server. The DSC Module allows you to connect tags and data directly through the Logos networking protocol and can act as a Logos server.

The DSC Module also adds OPC functionality to LabVIEW, allowing DSC Module tags to connect to OPC servers and clients. Again, you do not need to perform any special configuration operations to use this OPC connectivity with the Tag Configuration Editor and the Tag Configuration Wizard, you can browse for any OPC servers on the network and access those tags.

Registering Network Computers

To access LabVIEW applications or FieldPoint hardware using the Logos networking protocol, you must register the FieldPoint device or the computer running an application. Register computers through the Tag Configuration Editor, the Tag Monitor, the DSC Module Options dialog box, or the Server Browser.

Note Some computers 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** from your application).
2. Select **Tools»DSC Module»Options**, click the **Advanced** tab, and click the **Registered Computers** button. Or open one of the following utilities:
   - Tag Configuration Editor (**Tools»DSC Module»Configure Tags**), then select **Configure»Network»Registered Computers**.
   - Tag Monitor (**Tools»DSC Module»Monitor Tags**), right-click **Network Neighborhood**, and select **Register Computer** from the shortcut menu.
   - Server Browser (**Tools»DSC Module»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 Computers** dialog box. The **Register Computer** dialog box appears.

b. Enter the name of the computer you want to access in the **Computer Name** text box, or browse for the computer in the network tree.

c. Click the **Register** button, then click the **OK** button when you finish.

4. Complete the following steps to remove a computer from the list of registered computers.

a. In the **Registered Computers** dialog box, select the computer you want to remove.

b. Click the **Remove** button, then click the **OK** button.

You also can unregister a computer in the Tag Monitor by right-clicking it and selecting **Unregister Computer** from the shortcut menu.

Setting up Time Synchronization for Network Computers

To keep data properly time stamped, make sure the times on the computers are properly synchronized. Select **Tools»DSC Module»Options** from your application to open the **Options** dialog box. On the **Advanced** tab, click the **Time Synchronization** button to configure time synchronization. The National Instruments time synchronization service is installed as a service in Windows 2000/NT/XP that runs every time you run a computer. Time synchronization runs as a background process in Windows Me/98.

Determining Time Server Search Order

Suppose you have four computers you need to have synchronized. Choose a primary time synchronization server and a backup. Make sure that the order of search for time servers is the same for all the computers on the network that 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/98 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 7-1.

Table 7-1. 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 synchronizes to itself. Computer B synchronizes to Computer A, if A is running. If A is not running, B synchronizes to itself. Computer C synchronizes to Computer A if it is running, Computer B if A is not running, and to itself if neither A nor B is running. Use this pattern for all the computers you want in one synchronized set.

Configuring Time Synchronization

Complete the following steps to configure time synchronization.

1. Select **Tools»DSC Module»Options** from your application.
2. Click the **Advanced** tab.
3. Click the **Time Synchronization** button. The **Time Synchronization Settings** 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** list. If no computer is set as a primary time server, the computer synchronizes to itself.
4. To add a computer to the **Time Server Search Order** list, click the **Add** button. If you know the name of the computer you want to add, you can type it into the **Host** textbox. If you do not know the exact name of the computer, you can browse for it in the **Select Computer** network tree.

   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 computers search for a time synchronization server, select the computer name and click the **Up** or **Down** button.
   - If you have some computers running Windows Me/98 and other computers running Windows 2000/NT/XP in the network, list

Windows 2000/NT/XP computers first in the server search list. Time synchronization works better between Windows Me/98 and Windows 2000/NT/XP systems when the Windows 2000/NT/XP computer is the server.

- When a computer you are configuring time synchronization for is running LabVIEW, you do not need to add that computer to the **Time Server Search Order** list.

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 the 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 Network Computers

For user accounts to work consistently across the network, you must use the same `lookout.sec` file for installed copies of the DSC Module. After you create the `lookout.sec` file, make a copy of it and place the copy in the Windows system directory of each computer you want to be able to use these user accounts with.

Monitoring Services

The National Instruments networking protocol requires several 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` (Citadel 4), `nicitdl5.exe` and `nidscsm.exe` (Citadel 5), `lkads.exe`, and `lktsrv.exe`.

Note The DSC Module also installs the Microsoft SQL Server Desktop Engine, which appears as `msde.exe` in the Services utility. You must run this service to log alarms.

Caution Do *not* stop these services while the DSC Module or the Tag Engine is running.

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**.

Viewing Client Connections

To see which computers are currently accessing data from a DSC Module application, use the Engine Manager. Refer to the *Viewing Tag Engine Status* section in Chapter 3, *Using Tags to Manage Input/Output in LabVIEW*, for more information about the Engine Manager.

Refer to the *LabVIEW Datalogging and Supervisory Control Module Run-Time System Help* for more information about viewing client connections.

Troubleshooting Communication Problems

If you are having communication problems over a network, review the following guidelines.

- Make sure each network computer can access the others across the network.
- Because the protocol used for network communication between network Tag Engines is based on TCP/IP, make sure each computer has TCP/IP configured correctly on it.
- Each computer must have a unique IP address and a host name assigned to it. You can use TCP/IP utilities such as `ping` (all operating systems) and `nslookup` (only Windows 2000/NT/XP) 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 network computers are separated by a firewall, you will have to make some adjustments. For more information about networking across firewalls, refer to the NI Developer Zone at `ni.com/zone`.
- Use Tag Monitor to monitor communication and determine whether data is accessible through Logos.

Configuring Startup VIs

Startup VIs are VIs that run automatically when LabVIEW starts. Complete the following steps to configure startup VIs.

1. Select **Tools»DSC Module»Advanced»Startup VIs** from your application to display the **Configure Startup VIs** dialog box.
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.
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 a Citadel Database

This appendix describes Structured Query Language (SQL), Open Database Connectivity (ODBC), and accessing Citadel database data using both SQL and ODBC.

Introduction

The Citadel historical database includes an ODBC driver, which enables other applications to directly retrieve data from a Citadel database using SQL queries.

The Citadel 5 ODBC driver is compliant with SQL 92 and ODBC 2.5 standards. All clients that follow these standards can retrieve data from a Citadel 5 database. ADO clients can now use the ODBC driver through the Microsoft OLE DB Provider for ODBC Drivers to access the Citadel 5 database. MS Query wizard can be used to build-up 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 a Citadel database. The Citadel ODBC driver also includes many built-in data transforms to simplify statistical analysis of retrieved data.

Creating a Citadel ODBC Data Source

An ODBC data source is created automatically when you install the DSC Module and the default **System DSN** is automatically created along with a new Citadel database with the same name as the database. If you remove a database from the ODBC connection, the corresponding DSN is removed too, unless you modified the DSN manually in the OCBC setup.

Complete the following steps to create a Citadel ODBC data source for use with the DSC Module.

1. Shut down all ODBC applications, such as databases, spreadsheets, word processors, and Microsoft Query, before you run the ODBC applet.
2. Click the Windows **Start** button and select **Settings»Control Panel**.
3. Run the ODBC applet. It might be called **ODBC**, **ODBC Data Sources**, or something similar depending on your operating system.
4. Click the **User DSN** tab or the **System DSN** tab, depending on which type of data source you want to create. User Data Source Names (DSNs) are only visible to the user who created them on the current computer. System DSNs are available to all users on the current computer. Click the **Add** button.
5. Select **National Instruments Citadel 5 Database**, then click the **Finish** button.
6. In the **National Instruments Citadel ODBC Setup** dialog box enter the **Data Source Name**, **Description**, and **Database URL** fields.
   - The **Data Source Name** is the name that ODBC applications use to select the data source and must be different from any other ODBC data source name.
   - **Description** is a free-form text string you can enter to describe the data source.
   - **Database URL** is the computer and database name for the data you intend to access, such as `\\computername\database_name`. Click the **...** browse button to browse to a local or remote database.
7. Click **Test Connection** to test the connection to the database.

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** value in the **National Instruments ODBC Setup** dialog box 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. The DSC Module generates aliases using internal trace IDs for all traces whose names exceed the **Maximum Column Name Length**. Refer to the *Aliases Table* section for more information about internal trace IDs.

8. Click **OK** in the **Setup** dialog box, then click **OK** in the **ODBC Data Source Administrator** dialog box.

Accessing Citadel Data

Access Citadel data using ODBC queries from data tables.

Aliases Table

Generated alias names are stored in table **Aliases**. The Aliases table has two columns: **AliasName** and **FullName**.

Database tables and column names can be only up 126 characters long and should not contain some special characters. Some ODBC clients support only names up to 62 characters long. Note that the database URL is not included in the trace or data set name.

The Citadel driver automatically generates alias names for tags (traces) or data sets whose names are longer than the **Maximum Column Name Length** value specified in the **National Instruments ODBC Setup** dialog box.

The alias name consists of a prefix and the original trace or data set name substring, so the total length of the alias string is equal to the **Maximum Column Name Length** value. The alias prefix has the following format *~XXXXXXXXXXXXXXXX_* where *XXXXXXXXXXXXXXXX* is a 64-bit trace or data set ID.

For example, if the original tag URL is `\\computername\my_database\my_process_folder\my_process\my_folder_1\my_folder_2\pot1.value` and **Maximum Column Name Length** is 32, the alias name would be `~A012ABC4045FE43A_r_2/pot1@value` where `A012ABC4045FE43A` is the internal trace ID. Note that database URL (`\\computername\my_database\`) is not used for the alias and that certain special characters were mapped to supported characters.

Some ODBC clients do not handle certain special characters in column and table names. Special characters in tag names and data set names are replaced as shown in Table A-1.

Table A-1. Special Access SQL Characters

Special Character	Converted Character
.	@
\	/

The special characters changed in ODBC 5. 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 the earlier processes.

IntData Table

With Citadel 5, the IntData table replaces the Traces table. The ODBC driver presents Citadel data to other applications as an IntData 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: **IntInterval**, **LocalTime**, and **UTCTime**.

Because Citadel is event-driven, it only logs a value when the value changes. Using **IntInterval**, you can query Citadel for values evenly spaced over a period of time. The Citadel service stores the time in **UTCTime** format and derives **LocalTime** from the stored time. The time zone is configured per database through the **National Instruments Citadel ODBC Setup** configuration dialog box.

Table A-2. IntData Table

Column Name	Description
LocalTime	Time stamps that indicate local time when values are logged. Note that local time is calculated from UTC time using current time zone setting.
UTCTime	Time stamps that indicate UTC time when values are logged.
IntInterval	Interpolation interval. Replaces **Interval** column available in 4.*x* version. Default interpolation interval is one day. Note that syntax of interpolation interval string was not changed and special interval types (**YEAR**, **MONTH**, **WEEK**) can be still used.
TagName	Tag name.

IntInterval specifies the query value sample rate and can range from 1 ns to several years. The default value of **IntInterval** is 1 (one day). **IntInterval** is displayed as a regular table column. Display format depends on the IntInterval value specified in the *where* clause. Fixed intervals are displayed as `hh:mm:ss.ffffff`. Special intervals WEEK, MONTH, YEAR are displayed in days.

Table A-3. IntInterval Example Values

IntInterval	Interval length	Note
1	One day (24 hours)	—
1.5	One and half day (36 hours)	—
‘0:2’	2 seconds	—
‘5:2.125’	5 minutes, 2 seconds, and 125 milliseconds	—
‘10:0:0.001’	10 hours and 1 millisecond	—
‘WEEK’	7 days	—
‘MONTH’	1 month	Accounts for different month lengths and leap years.
‘YEAR’	1 year	Accounts for different month lengths and leap years.

RawData Table

With Citadel 5, the RawData table replaces the Points table of Citadel 4. The RawData 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 time stamps for one tag and another. For this reason, when querying the RawData table, you can query only one tag at a time.

Table A-4. RawData Table

Column Name	Description
LocalTime	Time stamps that indicate local time when values are logged. Note that local time is calculated from the logged UTC time using current time zone setting.
UTCTime	Time stamps that indicate UTC time when values are logged.
LoggingTime	Time stamps that indicate local time when values are logged regardless of local time zone setting.
TagName	Tag name.
Quality	Tag quality.

The possible values for **Quality** are in Table A-5.

Table A-5. Possible Values for Quality

Value	Description
0x00000000	Quality good
0x00000001	Stale
0x00000002	Sensor Failure
0x00000004	Device Failure
0x00000008	Server Failure
0x00000010	Network Failure
0x00000020	Nonexistent
0x00000040	No Known Value
0x00000080	Inactive

Table A-5. Possible Values for Quality (Continued)

Value	Description
0x00000100	Forced
0x00000200	Low Limited
0x00000400	High Limited
0x00000800	Constant
0x00001000	Sensor Inaccurate
0x00002000	EU Limits Exceeded
0x00004000	Subnormal
0x00008000	Math Exception
0x00010000	Comm Link Failure

The *where* clause using **LocalTime**, **UTCTime**, and **LoggingTime** is supported for the RawData table. However, **IntInterval** is not relevant to the RawData table. The data transforms are also not relevant to the RawData table and are not supported. Note that standard set functions (MAX, MIN, AVG, COUNT) are supported.

Dataset Tables

IntData and **RawData** tables contain all tags available in given Citadel database. Data set tables contain only tags available in particular data sets. There are **DS_IntData**_*dataset_name* and **DS_RawData**_*dataset_name* tables per data set. In addition to IntData or RawData table columns, data set tables have **RunID** and **RunName** columns.

In addition to the example queries, the *where* clause can restrict query to specified data set run(s).

Table A-6. DS_IntData_dataset_name Table

Column Name	Description
LocalTime	Time stamps that indicate local time when values are logged. Note that local time is calculated from UTC time using current time zone setting.
UTCTime	Time stamps that indicate UTC time when values are logged.

Table A-6. DS_IntData_dataset_name Table (Continued)

Column Name	Description
IntInterval	Interpolation interval. Default interpolation interval is one day.
RunID	Unique 64-bit identification of data set run.
RunName	Name of data set run. Note that this name is not unique and can be NULL.
TagName	Tag name.

Table A-7. DS_RawData_dataset_name Table

Column Name	Description
LocalTime	Time stamps that indicate local time when values are logged. Note that local time is calculated from UTC time using current time zone setting.
UTCTime	Time stamps that indicate UTC time when values are logged.
LoggingTime	Time stamps that indicate local time when values are logged regardless of local time zone setting.
Quality	Tag quality.
RunID	Unique 64-bit identification of data set run.
RunName	Name of data set run. Note that this name is not unique and can be NULL.
TagName	Tag name.

Query Commands

Use data transform and tag type case commands to query the Citadel data tables.

Data Transforms

Queries can include special commands that perform data transforms to manipulate and analyze historical data. Data transform commands cannot be used to query raw data tables.

Table A-8. Data Transform Commands

Command	Transformation
MATH_MIN(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the minimum for tag across the interval.
MATH_MAX(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the maximum for tag across the interval.
MATH_AVG(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the average for tag across the interval.
MATH_STDEV(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the standard deviation for tag across the interval.
MATH_STARTS(tag | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the number of starts (that is, the number of transitions from OFF to ON) for tag across the interval. For numeric points, 0.0 is interpreted as OFF, and all other numbers are treated as ON.
MATH_STOPS(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the number of stops (that is, the number of transitions from ON to OFF) for tag across interval.
MATH_ETM(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	Returns the amount of time tag was in the ON state across the interval.
MATH_QUAL(*tag* | TO_DISCRETE(*tag*) | TO_CONTINUOUS(*tag*))	There might be gaps in the historical data traces in Citadel because of machine shutdown, Tag Engine shutdown, or similar occurrences. MATH_QUAL returns the ratio of time for which valid data exist for tag across the interval to the length of the interval itself. If valid data exist for only one-half of the interval, MATH_QUAL returns 0.5.

Note that standard **set functions** (MIN, MAX, AVG, COUNT) are supported and can be used on any Citadel table. It is important to distinguish between set functions MIN, MAX, AVG, and Citadel **transform functions** MATH_MIN, MATH_MAX, and MATH_AVG. Set functions perform calculations on query result. Transform functions perform calculations on interpolation intervals.

Tag Type Cast Commands

Type cast commands can be used to override current type of tag that is being queried. It is important to distinguish between discrete and continuous tags because different point interpolation is used for different tag types. Discrete points are interpolated using step interpolation. Continuous points are interpolated using linear interpolation. In some cases may be useful to override tag type in order to enforce step or linear interpolation.

Table A-9. Tag Type Cast Commands

Citadel 5 ODBC driver	Description
TO_DISCRETE(*tag*)	Cast current *tag* type to discrete (that is, the *tag* is treated as discrete)
TO_CONTINUOUS(*tag*)	Cast current *tag* type to continuous (that is, the *tag* is treated as continuous)

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.

SQL Examples

The following examples are typical query statements; however, queries might be much more involved, depending on your system requirements.

Aliases Table Example Queries

The following example queries obtain Citadel data from the Aliases table.

```
SELECT *
FROM Aliases
```

Retrieves all aliases.

```
SELECT *
FROM Aliases
WHERE FullName LIKE '%Process_1%'
```

Retrieves all aliases of **Process_1** traces.

```
SELECT *
FROM Aliases
WHERE AliasName LIKE 'DS_%'
```

Retrieves all data set aliases.

```
SELECT *
FROM Aliases
WHERE AliasName NOT LIKE 'DS_%'
```

Retrieves all trace aliases.

IntData Table Example Queries

The following example queries obtain Citadel data from the IntData table.

```
SELECT *
FROM IntData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-11-29 18:00:00'
AND IntInterval = '1:0'
```

Selects data over a specified time at one-minute intervals.

```
SELECT * FROM IntData
WHERE LocalTime BETWEEN '2001-11-29 17:00:00' AND
'2001-11-29 18:00:00'
AND IntInterval = '1:0'
ORDER BY "computername/my_process/pot1@value"
```

Selects data over a specified time at one-minute intervals and orders rows based on values in column `computername/my_process/pot1@value`.

```
SELECT * FROM IntData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00'
AND '2001-11-29 18:00:00'
AND IntInterval = '1:0'
ORDER BY "computername/my_process/pot1@value" DESC
```

Selects data over a specified time at one-minute intervals and orders rows based on values in column `computername/my_process/pot1@value` in descending order.

```
SELECT * FROM IntData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00'
AND '2001-11-29 18:00:00'
AND IntInterval = '1:0'
AND "computername/my_process/pot1@value"
BETWEEN 1.5 AND 5.4
ORDER BY "computername/my_process/pot1@value" DESC
```

Selects data over a specified time and value interval at one-minute intervals and orders rows based on values in column `computername/my_process/pot1@value` in descending order.

```
SELECT MIN("computername/my_process/pot1@value"),
MAX("computername/my_process/pot1@value"),
AVG("computername/my_process/pot1@value")
FROM IntData
WHERE UTCTime
BETWEEN '2001-11-29 17:00:00' AND '2001-11-29 18:00:00'
AND IntInterval = '1:0'
```

Selects data over a specified time at one-minute intervals and returns minimum, maximum and average value of the `computername/my_process/pot1@value` column.

RawData Table Example Queries

The following example queries obtain Citadel data from the RawData table.

```
SELECT LocalTime, UTCTime,
"computername/my_process/pot1@value"
FROM RawData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-12-01 17:00:00'
```

Selects the `computername\my_process\pot1.value` tag data over a specified time.

```
SELECT LocalTime, UTCTime,
"computername/my_process/pot1@value"
FROM RawData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-12-01 17:00:00'
AND NOT Quality = 0
```

Selects the `computername\my_process\pot1.value` tag data with bad quality over a specified time.

```
SELECT LocalTime, UTCTime,
"computername/my_process/pot1@value"
FROM RawData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-12-01 17:00:00'
ORDER BY "computername/my_process/pot1@value" DESC
```

Selects the `computername\my_process\pot1.value` tag data over a specified time and order result rows in descending order.

```
SELECT MIN("computername/my_process/pot1@value"),
MAX("computername/my_process/pot1@value")
FROM RawData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-12-01 17:00:00'
```

Selects the `computername\my_process\pot1.value` tag data over a specified time and search the minimum and maximum value.

Dataset Tables Example Queries

The following example queries obtain Citadel data from the Dataset tables.

```
SELECT LocalTime, UTCTime,
RunName,"computername/my_process/pot1@value"
FROM DS_RawData_My_Dataset
WHERE RunName = 'MyRun_1'
```

Selects the `computername\my_process\pot1.value` tag data over a data set run `MyRun_1`.

```
SELECT LocalTime, UTCTime, RunName,
"computername/my_process/pot1@value"
FROM DS_RawData_My_Dataset
WHERE RunName IN ('MyRun_1', 'MyRun_3')
```

Selects the computername\my_process\pot1.value tag data over a data set runs MyRun_1 and MyRun_3.

```
SELECT LocalTime, UTCTime, RunName,
"computername/my_process/pot1@value"
FROM DS_RawData_My_Dataset
WHERE RunName LIKE 'MyRun_%'
```

Selects the computername\my_process\pot1.value tag data over a data set whose names begin with string MyRun_.

```
SELECT Min(LocalTime), Max(LocalTime),
COUNT("computername/my_process/pot1@value")
FROM DS_RawData_My_Dataset
WHERE RunName = 'MyRun_3'
```

Queries for start time, end time and number of points of the MyRun_3 data set run.

Data Transform and Type Cast Command Example Queries

The following example queries obtain Citadel data using data transform and type cast commands.

```
SELECT LocalTime,
TO_DISCRETE("computername/my_process/pot1@value") FROM
IntData
WHERE LocalTime
BETWEEN '2001-11-29 17:00:00' AND '2001-11-29 18:00:00'
AND IntInterval = '1:0'
```

Selects the computername\my_process\pot1.value tag data over a specified time at one-minute intervals and tread tag as discrete.

```
SELECT LocalTime,
MATH_MIN("computername/my_process/pot1@value"),
MATH_MAX("computername/my_process/pot1@value") FROM
IntData
WHERE LocalTime
BETWEEN '1999-03-06' AND '1999-03-13'
AND IntInterval = '1.0'
```

Selects the `computername\my_process\pot1.value` tag data over a specified time at one-day intervals and return minimum and maximum day values.

```
SELECT LocalTime,
MATH_MIN("computername/my_process/pot1@value") AS
'min_value' FROM IntData
WHERE LocalTime
BETWEEN '1999-03-06' AND '1999-03-13'
AND IntInterval = '1.0'
ORDER BY min_value DESC
```

Selects the `computername\my_process\pot1.value` tag data over a specified time at one-day intervals and return minimum day values and order results in descending order.

```
SELECT LocalTime,
MATH_MIN(TO_DISCRETE("computername/my_process/pot1@valu
e"))
WHERE LocalTime
BETWEEN '1999-03-06' AND '1999-03-13'
AND IntInterval = 'WEEK'
```

Selects the `computername\my_process\pot1.value` tag data over a specified time at WEEK intervals and return minimum week values. Note that the `computername\my_process\pot1.value` tag is treated as a discrete tag.

Accessing Citadel Data from Other Software

Refer to NI Developer Zone, at `ni.com/zone` for information about using Citadel historical databases with other software such as Microsoft Excel, Microsoft Access, or Visual Basic. Use a search phrase such as `Citadel SQL`.

B

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/support. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/custed for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Symbol	Prefix	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 a *Human Machine Interface (HMI)*.

ACK (Acknowledge) — The sequence action that indicates recognition of a new alarm.

alarm — An abnormal process condition. In the LabVIEW Datalogging and Supervisory Control (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 input/output 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 DSC Module development system and run in the DSC Module Run-Time System environment.

B

bit array tag — A multibit value representation of a connection to a real-world input/output point or memory variable. In the DSC Module, this type of tag can be made up of up to 32 discrete values.

C

Citadel — National Instruments proprietary historical database.

D

DAQ	Data Acquisition.
data set	A group of tag values logged together as a set during a specified period of time.
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 Application Programming Interface (API) to a number of different communication protocols.
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 input/output 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 input/output point. In the 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 DSC Module application. If a locally-defined tag in the Tag Engine attempts to access tag data across the network, the local Tag Engine uses the engine user account.

engineering units (EU)	Terms of data measurement, such as degrees Celsius, pounds, or grams.
event	A user-defined condition that a tag can reach, including going in or out of alarm state, or the user setting a value for the tag.

G

group	*See* tag group or input/output (I/O) group.

H

historical trend	A graph of data showing values that were logged to disk.
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 DSC Module system.

I

ID tag (data set)	Denotes a particular data set and the time during which the data set run took place when logging data sets.
input tag	A tag that accepts Tag Engine values from a device server.
input/output (I/O) group	A set of related server items, all of which share the same server update rate and deadband.
input/output tag	A tag that accepts Tag Engine values from a device server and sends values to the server.
IP	Internet Protocol.
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 DSC Module development system.
LabVIEW Real-Time (RT) Module	LabVIEW Real-Time software.
log deadband	The range through which a tag value must change before it is logged to a Citadel historical database.
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.
MB	Megabytes of memory.
memory tag	A tag not connected to a real-world input/output 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 that 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 Tag Engine.

P

Panel Wizard	A utility in the DSC 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 input/output 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. 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	A user account that handles unrecognized access requests to your data.

R

real-time trend	A graph of data that is updated as each new point is acquired in the Tag Engine.
reentrant execution	Mode in which calls to multiple instances of a subVI can run in parallel with distinct and separate data storage.

S

s	Seconds.
sampling period	The time interval between observations in a periodic sampling control system.
SCADA	Supervisory Control and Data Acquisition.
.scf	A configuration file that stores tag information and Tag Engine parameters.
sensor	A device that produces a voltage or current output representative of some physical property being measured, such as speed, temperature, or flow.
shift register	Optional mechanism in loop structures used to pass a variable's value from one iteration of a loop to a subsequent iteration.
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 a Citadel historical database.
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 input/output 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 input/output 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 DSC Module Run-Time System.
system errors	Errors that happen in the DSC 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 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 input/output 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 all tag values and alarm states, running 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.
time stamp	The exact time and date at which a tag value was sampled. Tag values are stored with their time stamps in the Tag Engine.
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 Tag Engine.
URL	Uniform Resource Locator. A URL is the way the DSC Module locates data across a network.

V

V	Volts.
VI	Virtual Instrument. Program in LabVIEW that models the appearance and function of a physical instrument.

Index

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W