

RELEASE AND UPGRADE NOTES

LabVIEW™ Datalogging and Supervisory Control Module

Version 8.2

This document describes the system requirements and the process of installing and upgrading the LabVIEW Datalogging and Supervisory Control (DSC) Module to version 8.2. This document also includes the new features available with version 8.2.

Because the DSC Module 8.x is significantly different from the DSC Module 7.x, this document also includes the differences between version 8.0 and previous versions of the DSC Module.

Refer to the *Getting Started with the LabVIEW Datalogging and Supervisory Control Module* manual for exercises you can complete to familiarize yourself with the new features and enhancements to the DSC Module.

Contents

| | |
|--|----|
| System Requirements..... | 2 |
| Installing the LabVIEW DSC Module..... | 2 |
| New Features in the LabVIEW DSC Module 8.2..... | 3 |
| Printing Alarms and Events..... | 3 |
| Programmatically Creating and Configuring I/O Servers..... | 4 |
| Upgrading from the LabVIEW DSC Module 7.x..... | 4 |
| New Features in the LabVIEW DSC Module 8.0..... | 4 |
| Configuring Shared Variables..... | 4 |
| Editing Multiple Shared Variables..... | 5 |
| Enabling Enhanced Security..... | 5 |
| Using Alarms..... | 7 |
| Configuring I/O Servers..... | 8 |
| Logging Data Directly to Citadel..... | 10 |
| Displaying Live and Historical Data..... | 10 |
| Front-Panel Binding..... | 10 |

| | |
|--|----|
| Binding Alarms to Data..... | 11 |
| Logging Arbitrary Data Types | 11 |
| Monitoring Data with New Event-Based PSP VIs..... | 11 |
| Using New Timeout Input for Historical VIs..... | 11 |
| Deleting Archived Alarms Using the Alarm & Event Archive VI | 11 |
| Changes to the DSC Terminology and Documentation | 12 |
| Terminology Changes | 12 |
| Documentation Changes | 13 |
| Upgrade or Compatibility Issues | 14 |
| VI Compatibility | 14 |
| Migrating the .scf File | 21 |
| Migrating Front Panel Security Settings | 22 |
| Citadel Database Compatibility | 23 |
| Converting Existing VI-Based Servers | 24 |
| Updating OPC Core Components | 24 |
| Known Issues..... | 25 |

System Requirements

To use the DSC Module, your computer must meet the following minimum system requirements:

- Windows 2000/XP.
- Minimum of 800 MB of additional free hard disk space to install the DSC Module and to have swap space available. You can increase computer resources to increase performance of DSC Module applications.
- Minimum of 512 MB of RAM minimum, 1 GB RAM recommended.
- LabVIEW Base, Full, or Professional Development System version 8.2. Refer to the *LabVIEW Release Notes* for LabVIEW system requirements.

Installing the LabVIEW DSC Module

Complete the following steps to install the DSC Module.

1. Log in to the development computer as an administrator or as a user with administrative privileges.
2. Install LabVIEW 8.2 from the LabVIEW 8.2 installation CD. Refer to the *LabVIEW Release Notes* for information about installing the LabVIEW development system.
3. Install the DSC Module from the LabVIEW DSC Module 8.2 installation CD. Follow the instructions that appear on the screen.

The DSC Module installs program files, documentation, and examples.



Note By default, the NI Keyboard Filter Driver is not installed. The NI Keyboard Filter Driver activates special security features, including the ability to restrict users from switching between applications by pressing the <Alt-Tab> keys. This driver does not work on laptop computers or on computers with hibernation enabled.

4. Restart the computer.

New Features in the LabVIEW DSC Module 8.2

The following sections describe the new features in the DSC Module 8.2. The [Upgrading from the LabVIEW DSC Module 7.x](#) section describes the changes between the LabVIEW DSC Module 7.x and 8.0. If you are upgrading from version 8.0, you only need to read this section. If you are upgrading from an earlier version of the DSC Module, you need to read this section as well as the [Upgrading from the LabVIEW DSC Module 7.x](#) section.

Printing Alarms and Events

Use the Alarm Printer I/O server to print alarms and events that LabVIEW processes generate. You can print all available information about an alarm or event, or you can select the information you want to print.

Right-click **My Computer** or a project library name in the Project Explorer window and select **New»I/O Server** from the shortcut menu to display the **Create New I/O Server** dialog box. Select **Alarm Printer** from the **I/O Server Type** list and click the **Continue** button to display the **Configure Alarm Printer Instance** dialog box. After you configure and deploy this I/O server, LabVIEW prints any alarms or events that LabVIEW processes generate to the printer you specify.

Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for more information about printing alarm and event data.

Programmatically Creating and Configuring I/O Servers

Use the Create and Configure IO Server Express VI to create or configure an I/O server programmatically. You can use this VI to create or configure a Modbus, ModbusSlave, or OPC I/O server. The configuration dialog boxes you use to configure the I/O servers are the same as those that appear when you create an I/O server from the **Project Explorer** window. However, you also can use the Create and Configure IO Server Express VI to modify the configuration of an I/O server at run-time.

Use the Create and Configure IO Server Express VI with the Create Process and the Create Shared Variable VIs to create or configure a process without using a LabVIEW library.

Use the Delete IO Server VI to delete an I/O server programmatically.

Refer to the *LabVIEW Help* for more information about creating and deleting Modbus, ModbusSlave, and OPC I/O servers.

Upgrading from the LabVIEW DSC Module 7.x

LabVIEW DSC Module 8.x significantly differs from LabVIEW DSC Module 7.x. Therefore, this section includes the upgrade information for the DSC Module 8.0, including the new features, changes to the terminology and documentation, and the upgrade or compatibility issues.

New Features in the LabVIEW DSC Module 8.0

This section describes the new features available in the DSC Module 8.0. Refer to the *LabVIEW Help* for more information about these new features.

Configuring Shared Variables

In previous versions of the DSC Module, you use tags to manage data in an application. In the DSC Module 8.0, you use shared variables to manage data in an application. The DSC Module adds data logging, alarming, security, and scaling to the shared variable.



Note You can configure shared variables as single-process or network-published. You can use the features the DSC Module adds only for network-published shared variables.

LabVIEW uses the NI Publish-Subscribe Protocol (NI-PSP) data transfer protocol to write and allow users to read live data across a network. NI-PSP replaces Logos from earlier versions of the DSC Module.

In previous versions of the DSC Module, you use the Tag Configuration Editor to configure tags. In the DSC Module 8.0, you can configure shared

variables through the LabVIEW project. To configure a shared variable, right-click the shared variable in the **Project Explorer** window and select **Properties** from the shortcut menu.



You can use the Shared Variable Constant, shown at left, and the SharedVariableIO properties and methods to configure shared variables and shared variable properties. To view the available properties and methods, place a Shared Variable Constant on the block diagram. Right-click the Shared Variable Constant and select **Create»Property** from the shortcut menu. The Property Node menu contains available properties.

Refer to the *LabVIEW Help* for more information about NI-PSP and about creating and managing shared variables.

Editing Multiple Shared Variables

Use the **Multiple Variable Editor** window to create, delete, and modify a large number of shared variables at one time. Select **Tools»Shared Variable»Multiple Variable Editor** to launch the **Multiple Variable Editor** window.

Use the **Multiple Variable Editor** window to export shared variable configuration information to spreadsheet files and import shared variable configuration information from spreadsheet files. The files are tab-delimited text (.CSV) files that you can open with a spreadsheet program, such as Microsoft Excel.

Enabling Enhanced Security

The DSC Module includes the following changes to security:

- **Front Panel Security**—Use the **Front Panel Security** dialog box to configure security for multiple controls or indicators at one time or to copy security settings from one control or indicator to another. Select **Tools»Security»Front Panel Security** from the front panel to access the **Front Panel Security** dialog box. Refer to the *Configuring Front Panel Security* section of the *Getting Started with the LabVIEW Datalogging and Supervisory Control Module* manual for more information about configuring security for front panel controls and indicators.
- **Security for Front Panel Objects**—In previous versions of the DSC Module, you configure security for front panel controls and indicators by using the Security Wizard, available by right-clicking a control or indicator and selecting **Security** from the shortcut menu. When you configure security, LabVIEW adds the NI SEC UI Security VI to the block diagram. Now, you can configure security by right-clicking a control or indicator, selecting **Properties** from the shortcut menu, and

selecting the **Security** tab. You can set access levels for users and groups for the front panel object on the **Security** tab. LabVIEW stores security settings with each control and does not add the deprecated NI SEC UI Security VI to the block diagram. Because LabVIEW stores security settings with each control, copying a control also copies the security settings.

LabVIEW applies the most restrictive security mode for the control or indicator. For example, if you configure security for a control on the **Security** tab and you also use a Property Node to affect the behavior of the same control, LabVIEW applies the more restrictive security mode. LabVIEW maintains both pieces of security information and applies them as necessary.

When you configure security for controls or indicators, the block diagram no longer contains the security code. Existing Security Wizard code is not compatible with the new security system. You must configure security settings on controls again when you upgrade existing VIs. You also can use the **Front Panel Security** dialog box to update the security settings automatically.



Note You cannot add security to ActiveX and .NET controls.

- **Upgraded Security System**—The security system now uses a security domain server. The new security system does not communicate with the User Account Manager that shipped with the DSC Module 7.1. The User Account Manager is no longer available with the DSC Module. Use the **Domain Manager** window, available by selecting **Tools»Security»Domain Account Manager**, to create and edit users and groups.

With the new security domain, you can share users and groups across multiple computers without replicating the User Account Manager. You can import the list of users and groups from the existing User Account Manager into the new **Domain Manager** window, but LabVIEW resets the passwords to empty passwords because of the differences in password encoding. You must reset passwords for all users and groups.

With the security domain, the group and user information cluster contains different content than the 7.1 version. All Security VIs now contain a **domain id** input to identify the domain server to which they connect.

Refer to the *LabVIEW Help* for information about configuring security settings for the LabVIEW environment.

Using Alarms

The DSC Module includes the following changes to alarming:

- **New Alarms & Events VIs**—The DSC Module 8.0 adds the Read Alarms VI, Read Events VI, Acknowledge Alarms VI, and Alarm Status VI. Each of the new VIs includes error handling.
 - Use the Read Alarms VI, which replaces the Read Tag Alarm VI, to obtain alarm information for one or more shared variables. The **alarm occurrences** output returns more information about the alarm than the Read Tag Alarm VI.
 - Use the Read Events VI, which replaces the Read Event History VI, to obtain information about events.
 - Use the Acknowledge Alarms VI, which replaces the Acknowledge Alarm VI, to acknowledge alarms by shared variable name, by instance, or by alarm area.
 - Use the Alarm Status VI, which replaces the Get Alarm Summary Status VI, to obtain status data about alarms.

In previous versions of the DSC Module, you can associate only one alarm with each tag. Now, you can create multiple alarms for each shared variable. Existing alarm states, such as LO, LO_LO, HI, HI_HI, and bad status, are now separate and distinct alarms. Each of these distinct alarms has two associated Boolean values—in alarm and acknowledged.

- **Set User-Defined Alarms and Events**—The DSC Module 8.0 adds the Set User Defined Event VI, Set User Defined Alarm VI, and Clear User Defined Alarm VI. In previous versions of the DSC Module, you must create a tag for each possible abnormal condition. With these VIs, you have greater control over reporting abnormal conditions in an application and no longer need to create a tag or shared variable. You also can decide between using an alarm or event for the abnormal condition.

Because user-defined alarms and events are not associated with a shared variable, you cannot read user-defined alarms or events using the alarm compatibility code.

- **Remote Alarms**—In previous versions of the DSC Module, you create a tag, which mirrors the remote system, to monitor remote alarms using the Read Tag Alarm VI. Now, you can monitor alarms over the network directly, which results in obtaining faster and more reliable alarm information.

- **Alarm Priorities**—You now can assign alarm priorities in the range 1–1,000 because the new Shared Variable Engine supports the complete range. Alarm priorities logged by the DSC Module 6.x and earlier supported the range 1–15. Citadel 4 originally supported Lookout, whose alarm priorities ranged from 1–10. Therefore, the DSC Module 6.x converted alarm priorities from the 1–15 range to the 1–10 range prior to logging data to the Citadel database. Citadel 5 supports the full range from 1–1,000. When you convert previously logged alarm priorities to the Citadel 5 format, the DSC Module retains those values.

Configuring I/O Servers

Previous versions of the DSC Module provided the VI-Based Server Development Toolkit, which you used to create VI-based servers. In the DSC Module 8.0, you can create custom I/O servers instead. A custom I/O server is a server that you use when you have a uniquely configured or custom device that you want to use with the DSC Module. You can use a custom I/O server to simulate hardware or connect the DSC Module to hardware for which no device server exists.

Right-click **My Computer** in the **Project Explorer** window and select **New»I/O Server** from the shortcut menu to display the **Create New I/O Server** dialog box. You can create **Custom VI – On Input Change**, **Custom VI – Periodic**, **DataSet Marking**, **Modbus**, **ModbusSlave**, and **OPC I/O** servers.

Custom VI-Based I/O Servers

You can create two types of custom VI-based I/O servers: periodic I/O servers and on input change I/O servers. Create a periodic I/O server when you want a continuous update of a data item with a user-configured update rate. Create an on input change I/O server if you want the custom I/O server to update based on a value change event. A change in the value of the input of the server data item is the only event that triggers a VI-based I/O server to update.

The DSC Module provides two assistants to help you create custom I/O servers. Use the Custom VI - Based Server – Periodic Assistant to create a periodic I/O server. Use the Custom VI - Based Server – On Input Change Assistant to create an on input change I/O server. These assistants replace the VI-Based Server Development Toolkit.

Creating Data Sets with the DataSet Marking I/O Server

Use the DataSet Marking I/O server to create a data set, which is a group of shared variable values logged together during a finite time period. You can use arbitrary NI-PSP data items for identifying the data set, the start time, and the stop time for logging the data. You also can use the associated NI-PSP data items to dynamically reconfigure the server during execution.

The data set includes an internal identification value that is now a human readable unique identifier that is identical to the identification value displayed in the Historical Data Viewer for each individual data set run.

Creating Modbus I/O Servers

In previous versions of the DSC Module, VIs communicated with Modbus devices through OPC servers. You now can create Modbus I/O servers that communicate directly with Modbus devices. You also can create Modbus slave I/O servers that act as Modbus slave devices. Both Modbus and Modbus slave I/O servers support serial port and Ethernet TCP communication. You can set VIs to read and write data to Modbus devices using shared variables.

OPC Client Enhancements

In the LabVIEW DSC Module 8.0, you can create an OPC client instance to connect to OPC servers. The DSC Module adds performance improvements when connecting to OPC servers and is compliant with versions 2.x and 3.0 of the OPC Data Access specification, as defined by the OPC Foundation. With versions 2.0 and 3.0 of the OPC Data Access specification, array data types are available.

You can set the reconnect poll rate for OPC client instances that connect to OPC servers. You also can set an OPC client instance to be active or inactive by binding a shared variable to the active OPC server data item.

To create the OPC client instance through the **Project Explorer** window, right-click a project library or **My Computer** and select **New»I/O Server** from the shortcut menu to display the **Create New I/O Server** dialog box. Select **OPC** from the **I/O Server Type** list and click the **Continue** button to display the **Configure OPC Client Instance** dialog box.

You can use the **Multiple Variable Editor** window to configure shared variables connected to the OPC server. In the **Multiple Variable Editor** window, select **Edit»Create Bindings** to access the **Create Bound Variables** dialog box. Use this dialog box to create shared variables that are bound to multiple OPC I/O points or ranges of I/O points.

Logging Data Directly to Citadel

You can use the new Database Writing VIs to write data directly to the Citadel historical database without configuring and running a shared variable. You can log data not automatically logged by the Shared Variable Engine or import data from an external data source to the Citadel database. Refer to the *LabVIEW Help* for more information about using the Database Writing VIs.

Displaying Live and Historical Data

Use the Historical Trend Express VI to configure and display historical data from a Citadel database. The DSC Module does not replace existing Historical Trend Wizard code with the new Historical Trend Express VI automatically. The configuration page for existing Historical Trend Wizard code has been deprecated, so you cannot reconfigure the wizard code. Refer to the *Historical Trend* topic in the *LabVIEW Help* for more information about using the Historical Trend Express VI.

Use the Real-Time Trend Express VI to configure and display live data from a shared variable. The DSC Module does not replace existing Real-Time Wizard code with the new Real-Time Trend Express VI automatically. The configuration page for existing Real-Time Trend Wizard code has been deprecated, so you cannot reconfigure the wizard code. Refer to the *Real-Time Trend* topic in the *LabVIEW Help* for more information about using the Real-Time Trend Express VI.

You do not need to replace Historical Trend Wizard code or Real-Time Wizard code with the Historical Trend Express VI or Real-Time Trend Express VI unless you need to reconfigure the wizard code. If you need to reconfigure the wizard code, replace the code with the appropriate Express VI.

Front-Panel Binding

You can bind front panel controls and indicators to shared variables on the **Data Binding** tab of the control or indicator **Properties** dialog box, available by right-clicking the control or indicator and selecting **Properties** from the shortcut menu. You also can bind front panel controls and indicators to shared variables using the **Front Panel Binding Mass Configuration** dialog box, available by selecting **Tools»Shared Variable»Front Panel PSP Mass Configurator**.

Binding Alarms to Data

You can configure a control to blink when the data has an alarm. Place a checkmark in the **Blink when in Alarm** checkbox on the **Data Binding** tab of a property page. This option causes the control to blink continuously as long as the data has an alarm. To turn the blinking off manually, set the control property node blinking to FALSE. Existing Blink on Alarm Wizard code is not replaced by the new property value. You must use the new **Blink when in Alarm** checkbox. The DSC Module no longer supports setting the color of a control or indicator based on the alarm condition.

Logging Arbitrary Data Types

You can log any type of data to Citadel. LabVIEW retrieves data using the variant data type.

Monitoring Data with New Event-Based PSP VIs

The DSC Module now includes Event Structure Support VIs to monitor shared variables and associated Alarm & Event occurrences using the Event structure. You can receive data update notifications directly from the Event structure in the main loop of the application without complicated polling-based block diagram code for reading shared variable data.

Using New Timeout Input for Historical VIs

The following VIs now contain a **timeout (30 s)** input: Write Traces to Spreadsheet, Write Traces to Spreadsheet File, Detach Database, Delete Traces, Compact Database, Archive Traces, Alarm and Event Archive, Alarm & Event Query To Spreadsheet File, Create Database, Get Trace Info, Get Trace List, and Rename Historical Data. The **timeout (30 s)** input limits the time that the VI waits for the operation to complete. The default value is 30 seconds. A timeout value of less than or equal to 0 indicates no timeout, and the VI waits indefinitely until the data is returned. The VI ignores the **timeout (30 s)** input if you set **show status bar** to TRUE.

Deleting Archived Alarms Using the Alarm & Event Archive VI

The Alarm & Event Archive VI now contains a **delete data from source database** input. Set this input to TRUE to delete archived alarms from the source database. You can delete data only from a local Citadel database. If you try to delete data from a remote database, the VI returns an error.

When you want to delete data from the database, make sure that LabVIEW is not currently using the database. The VI returns an error if the database is in use and you attempt to delete data.

Changes to the DSC Terminology and Documentation

This section contains changes to terminology and documentation in the DSC Module.

Terminology Changes

The following table describes the terms used in the DSC Module 7.1 and earlier and the terms used in the DSC Module 8.0.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 | Definition |
|---|-------------------------------|--|
| tag | shared variable | A shared variable represents a value or an I/O point. You can use a shared variable to access and pass data among several VIs across a network. You also can share data across VIs that are not in the same project. The shared variable is available without the DSC Module. However, with the DSC Module, you can configure logging, deadband, alarming, security, and scaling for shared variables. You can use the features the DSC Module adds only for network-published shared variables. |
| memory tag | shared variable | In previous versions of the DSC Module, you used memory tags to represent values not connected to a real-world I/O point. You can use a shared variable for the same purpose in the DSC Module 8.0. |
| Tag Engine | Shared Variable Engine | The Shared Variable Engine maintains all shared variable values and alarm states running as a process independent of the HMI application. The Shared Variable Engine starts automatically when you log into Windows if you have the DSC Module installed. Therefore, you do not need to start and stop the Shared Variable Engine. |
| Tag Monitor | Published Variable Monitor | The Published Variable Monitor allows you to verify the network status of shared variables. |

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 | Definition |
|---------------------------------------|----------------------------------|--|
| tag attributes | shared variable properties | Shared variable properties pertain to a shared variable. Properties include alarming options, security options, logging options, and engineering units. You can configure items by using the SharedVariableIO properties and methods. Refer to the Configuring Shared Variables section of this document for more information. |
| tag group | alarm area | An alarm area consists of a group of alarms. An alarm area enables you to sort alarms and to acknowledge a group of alarms. |
| .scf file | LabVIEW project library (.lvlib) | The LabVIEW project library contains references to specific shared variables and allows you to set properties and access information about shared variables. Refer to the Migrating the .scf File section of this document for information about converting .scf files to .lvlib files. |
| Data Set Logger | DataSet Marking I/O Server | Use the DataSet Marking I/O server to create a data set, which is a group of shared variable values logged together during a finite time period. Refer to the Creating Data Sets with the DataSet Marking I/O Server section of this document for more information about the DataSet Marking I/O server. |

Documentation Changes

All documentation for the DSC Module 8.0, except this document and the *Getting Started with the LabVIEW Datalogging and Supervisory Control Module* manual, appears in the *LabVIEW Help*. To access the *LabVIEW Help*, select **Help»Search the LabVIEW Help**. The following table lists manuals from previous releases and the location where you can find the information for the DSC Module 8.0.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|--|---|
| <i>LabVIEW DSC Module Bookshelf</i> | This document is no longer available. You can search DSC Module documentation using the Search tab in the LabVIEW Help window. |
| <i>LabVIEW Datalogging and Supervisory Control Module Developer Manual</i> | You can find this content in the <i>LabVIEW Help</i> . Click the Contents tab in the LabVIEW Help window and select the DSC Module» DSC Module Concepts book. |
| <i>VI-Based Server Development Toolkit Reference Manual</i> | You can create a VI-based server using a custom VI I/O server assistant, which guides you through the process step-by-step. Refer to the Configuring I/O Servers section of this document for more information about creating VI-based servers. |

You can continue to use the *Getting Started with the LabVIEW Datalogging and Supervisory Control Module* manual to familiarize yourself with the new features and enhancements to the DSC Module.

Upgrade or Compatibility Issues

This section describes upgrade and compatibility issues for the DSC Module 8.0.

VI Compatibility

When you open a VI saved in a previous version of LabVIEW, LabVIEW automatically converts and compiles the VIs and prompts you to save. If you save a VI in the DSC Module 8.0, you no longer can load the VI in an earlier version. You can select **File»Save for Previous Version** to save a copy of the VI for use in LabVIEW 7.1.

This version of the DSC Module deprecates the following VIs and functions. Deprecated VIs and functions appear on the block diagram with a red outline around them. All of the deprecated VIs are located in the `vi.lib\lvdsc\legacy` folder and are unlocked so you can look at the implementation of them. National Instruments recommends that when you build new applications, you use the new VIs provided by the DSC Module 8.0.



Note When you open a VI that contains deprecated VIs, the VI might be broken because some of the inputs and outputs of the deprecated VIs were removed. These inputs and outputs were removed because the DSC Module 8.0 no longer supports that specific functionality.

The following sections list the deprecated VIs and functions and the new VIs that you can use to achieve the same functionality.

Alarms & Events VIs

The following table lists the Alarms & Events VIs that are deprecated and shows which VI or control you can use in the DSC Module 8.0. You can access the functionality from these VIs with updated Alarms & Events VIs.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---|---|
| Read Tag Alarm VI | <p>Use the Read Alarms VI.</p> <p>Enter an array of shared variable names. The Read Alarms VI returns an array of alarm occurrences associated with those shared variables.</p> |
| Acknowledge Alarm VI | <p>Use the Acknowledge Alarms VI.</p> <p>The Acknowledge Alarms VI is polymorphic. You can simultaneously acknowledge all alarm occurrences for a given shared variable or acknowledge specific occurrences for an alarm for a given shared variable.</p> |
| Read Alarm Summary VI | <p>Use the Read Alarms VI or the Alarm and Event Display control.</p> <p>The Read Alarms VI provides the same functionality as this deprecated VI. The Read Alarms VI returns the alarm information as a cluster of information. Use the Alarm and Event Display control to display this information in a table that is easier to read.</p> |
| Read Event History VI | <p>Use the Read Events VI or the Alarm and Event Display control.</p> <p>The Read Events VI provides the same functionality as this deprecated VI. The Read Events VI returns the event information as a cluster of information. Use the Alarm and Event Display control to display this information in a table that is easier to read.</p> |
| Get Alarm Summary Status VI | <p>Use the Alarm Status VI.</p> <p>The Alarm Status VI provides the number of alarms and the number of shared variables associated with the alarms.</p> |

The Alarm & Event Archive VI, Alarm & Event Query VI, and the Alarm & Event Query To Spreadsheet File VI are still available. You can find these VIs on the **Historical** palette.

Refer to the [Using Alarms](#) section of this document for more information about the new VIs.

System VIs

The following table lists the System VIs that are deprecated and shows which VI or function you can use in the DSC Module 8.0. You can access most of the functionality from these VIs with the new Engine Control VIs.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---|--|
| Engine Launch VI | The DSC Module does not start or shut down the Shared Variable Engine. The Shared Variable Engine runs automatically. Use the Deploy Library VI to load a process into the Shared Variable Engine. |
| Engine Shutdown VI | The DSC Module does not start or shut down the Shared Variable Engine. The Shared Variable Engine runs automatically. Use the Undeploy Library VI to remove a process from the Shared Variable Engine. |
| Enable Event Logging VI | Use the Enable Alarm Logging On Process VI. |
| Enable Historical Data Logging VI | Use the Enable Data Logging On Process VI. |
| Enable Printing VI | The DSC Module no longer supports line printing of alarms and events. Refer to the National Instruments Web site at ni.com/info and enter the info code <code>excdwi</code> for information about printing. |
| Get Logging Status VI | Use the Get Process Log State VI. The logging config changed occurrence and printing enabled outputs of the Get Logging Status VI do not have equivalent properties in the DSC Module 8.0. When you open an existing VI that contains this VI, these outputs will not be connected and will have broken wires on the block diagram. The Get Logging Status VI in the <code>legacy</code> folder no longer contains these outputs. |
| Get Engine Status VI | The DSC Module does not start or shut down the Shared Variable Engine. The Shared Variable Engine runs automatically. Use the Get Process List VI and the Get Process Run State VI. |

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---|--|
| Get Tag Status Info VI | Use the DataSocket Read function to get an error code. |
| Tag Status Handler VI | This functionality is deprecated. |
| Post System Error or Event VI | Use the Set User Defined Event VI. |
| Load SCF VI | This functionality is deprecated. In previous versions of the DSC Module, you must load the <code>.scf</code> file that contains information about the tags you want to access. Use the Deploy Library VI to load the shared variables in a library into the Shared Variable Engine. Use the Create Process VI to create an empty process in the Shared Variable Engine. Refer to the Migrating the .scf File section of this document for information about converting an existing <code>.scf</code> file to a project library. |

Tags VIs

The following table lists the Tags VIs, functions, and constants that are deprecated and shows the VI, property, or function you can use in the DSC Module 8.0.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---|--|
| Read Tag VI | Use the DataSocket Read function. |
| Write Tag VI | Use the DataSocket Write function. |
| Read Multiple Tags VI | Use the DataSocket Read function inside a For Loop. |
| Write Multiple Tags VI | Use the DataSocket Write function inside a For Loop. |
| Trend Tags VI | Use the Historical Trend Express VI or Real-Time Trend Express VI. |
| Tag Constant | Use the Shared Variable Constant. |
| Not a Tag Constant | Use the Not a Shared Variable Constant. |
| String to Tag function | Use the String to Shared Variable function. |

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---------------------------------------|---|
| Tag to String function | Use the Shared Variable to String function. |
| Tag Name to DataSocket URL VI | Use the DataSocket URL SharedVariableIO property by right-clicking the shared variable on the block diagram and selecting Create»Property»DataSocket URL . You also can wire the Shared Variable Constant directly into a DataSocket VI or function. |

LabVIEW automatically converts Tag constants, Not a Tag constants, String to Tag functions, and Tag to String functions into the LabVIEW 8.0 constant or function when you open an existing VI.

If you have existing VIs that use the Start Tag Monitor VI, you can continue to use the VI.

Refer to the *LabVIEW Help* for information about using the DataSocket VI and functions to programmatically configure, read, or write live data using NI-PSP.

Tag Attributes VIs

The following table lists the Tag Attributes VIs that are deprecated and shows which VI or property you can use in the DSC Module 8.0. You can access the functionality from most of these VIs with the SharedVariableIO properties. To access the SharedVariableIO properties, right-click a Shared Variable Constant on the block diagram and select **Create»Property** from the shortcut menu.

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---------------------------------------|---|
| Get Tag Attribute VI | Use the SharedVariableIO properties available when you right-click a Shared Variable Constant and select Create»Property from the shortcut menu. |
| Set Tag Attribute VI | Use the SharedVariableIO properties available when you right-click a Shared Variable Constant and select Create»Property from the shortcut menu. To set shared variable properties, select Change To Write from the shortcut menu of the Property Node. You cannot set all properties. The Change To Write option appears dimmed if you cannot set the property. |

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---------------------------------------|--|
| Set Multiple Tag Attributes VI | In a For Loop, use the SharedVariableIO properties available when you right-click a Shared Variable Constant and select Create»Property from the shortcut menu. To set shared variable properties, select Change All To Write from the shortcut menu of the Property Node. You cannot set all properties. The Change To Write option appears dimmed if you cannot set the property. |
| Get Tag List VI | Use the Get Shared Variable List VI. |
| Get Group List VI | Use the Alarming: <i>AlarmName</i> :Area SharedVariableIO property, the Index Array function, and the Unbundle By Name function, where <i>AlarmName</i> is HI, HI_HI, LO, or LO_LO. A tag group is now an alarm area. The Alarming: <i>AlarmName</i> :Area SharedVariableIO property returns the alarm areas in the parent process. |
| Get Tag Description Group VI | Use the Data Type, Name, and Description SharedVariableIO properties, the Index Array function, and the Unbundle By Name function. A tag group is now an alarm area. The Description SharedVariableIO property returns the description of the alarm areas in the parent process. |
| Get Tag IO Connection Info VI | Use the Network:Access Type and Network:URL SharedVariableIO properties. |
| Get Tag Logging Info VI | Use the Logging:Enabled, Logging:Log Data, and Logging:Log Alarms and Events SharedVariableIO properties. |
| Get Tag Range and Units VI | Use the Scaling:Engineering Minimum, Scaling:Engineering Maximum, Scaling:Enabled, and Scaling:Units SharedVariableIO properties. |
| Tag Attribute Ring control | Use the SharedVariableIO properties. |
| Get Tag Alarm Enabled VI | Use the Alarming: <i>AlarmName</i> :Enabled and Alarming: <i>AlarmName</i> :Ack Type SharedVariableIO properties, where <i>AlarmName</i> is HI, HI_HI, LO, or LO_LO. |

| LabVIEW DSC Module 7.1 and Earlier | LabVIEW DSC Module 8.0 |
|---|--|
| Get Analog Tag Alarm Limit VI | Use the Name, Alarming:Alarms Enabled, Alarming:AlarmName:Enabled, Alarming:AlarmName:Level, Alarming:AlarmName:Deadband, Alarming:AlarmName:Priority, Alarming:AlarmName:Ack Type, Alarming:AlarmName:Area, Alarming:AlarmName:Description, and Alarming:AlarmName:Allow Log SharedVariableIO properties, where AlarmName is HI, HI_HI, LO, or LO_LO. |
| Get Discrete Tag Alarm Setting VI | Use the Name, Alarming:Alarms Enabled, Alarming:Boolean:Enabled, Alarming:Boolean:Alarm On, Alarming:Boolean:Priority, Alarming:Boolean:Ack Type, Alarming:Boolean:Area, Alarming:Boolean:Description, and Alarming:Boolean:Allow Log SharedVariableIO properties. |
| Get BitArrayTag Alarm Setting VI | Use the Name, Alarming:Alarms Enabled, Alarming:U32 Bit Field:Enabled, Alarming:U32 Bit Field:Invert Mask, Alarming:U32 Bit Field:Select Mask, Alarming:U32 Bit Field:Alarm On, Alarming:U32 Bit Field:Priority, Alarming:U32 Bit Field:Ack Type, Alarming:U32 Bit Field:Area, Alarming:U32 Bit Field:Description, and Alarming:U32 Bit Field:Allow Log SharedVariableIO properties. |
| Get Tag Bad Status Alarm Info VI | Use the Name, Alarming:Alarms Enabled, Alarming:Bad Status:Enabled, Alarming:Bad Status:Priority, Alarming:Bad Status:Ack Type, Alarming:Bad Status:Area, Alarming:Bad Status:Description, and Alarming:Bad Status:Allow Log SharedVariableIO properties. |

DSC Module Server Development VIs

All of the VIs on the **DSC Module Server Development** palette are deprecated. All of the controls on the **DSC Module Server Data Types** palette are deprecated. The DSC Module now provides the Custom VI Based Server – On Input Change Assistant and the Custom VI Based Server – Periodic Assistant to create custom VI-based I/O servers.

The deprecated VIs are available in the `vi.lib\lvds\legacy\svrlibs` folder. You can continue to use the deprecated VIs to create custom VI-based I/O servers if necessary. National Instruments recommends that you use one of the assistants for any new development.

Refer to the *Custom VI-Based I/O Servers* section of this document for more information about the custom VI-based I/O server assistants. Refer to the *Converting Existing VI-Based Servers* section of this document for more information about converting an existing VI-based server to a VI-based server you can use in the DSC Module 8.0.

Alarms and Events Controls Palette

The Alarm Summary Display Listbox, Event History Display Listbox, Alarm Summary Format, Event History Format, Color Codes for Event History, Color Codes for Alarm Summary, and Alarm & Event Query Data Format controls are deprecated.

Color coding is no longer available for alarms in the DSC Module. Because of format changes in the way that the DSC Module handles alarms, the Alarm Summary Display Listbox, Event History Display Listbox, Alarm Summary Format, Event History Format, and Alarm & Event Query Data Format controls are unavailable. You can arrange alarm information into a multicolumn listbox. Refer to the NI Example Finder, available by selecting **Help»Find Examples**, for examples that illustrate how to format alarm information.

Migrating the .scf File

In previous versions of the DSC Module, the `.scf` file contains information about tags and tag parameters. This information is now contained in a project library (`.lvlib`). Complete the following steps to convert an existing `.scf` file to a project library.



Note You can convert 2.x, 6.x, and 7.x versions of `.scf` files.

1. Select **Tools»DSC Module»Migrate»Configuration (.scf)** to launch the **SCF Migration** dialog box and the **Configuration (.scf) File Migration** dialog box.

2. In the **SCF Migration** dialog box, click the **Browse** button and navigate to the `.scf` file you want to convert.
3. Click the **Select** button.
4. In the **Configuration (.scf) File Migration** dialog box, click the **Convert** button to convert that `.scf` file to a project library.

If you select a `2.x` or `6.x` file, LabVIEW prompts you to convert that file into a `7.x` file first. You then can convert that file to a project library. LabVIEW also creates a backup of the `2.x` or `6.x` file. Click the **Convert** button.



Note If the `.scf` file you are converting is associated with Citadel 3.8 or `4.x`, the SCF Migration Assistant also prompts you to convert the database to version 5.0. Click the **Continue** button to convert the database.

5. LabVIEW creates a new `.lvlib` file and adds the tags from the `.scf` file to the project library. LabVIEW displays the location of the new project library. LabVIEW places the project library in the same location as the original `.scf` file.
6. Click the **Close** button.

Refer to the *LabVIEW Help* for information about the project library.

Migrating Front Panel Security Settings

To convert the front panel security settings for an existing VI or folder of VIs for use in the DSC Module 8.0, select **Tools»DSC Module»Migrate»Front Panel Security**. If you created a VI with the DSC Module 7.1 and earlier and that VI contains front panel security settings, you must use the **Front Panel Security** dialog box to convert the front panel security settings of that VI.

You can convert existing `lookout.sec` files for use in the DSC Module 8.0. To import existing security files, use the **Domain Manager** window, available by selecting **Tools»Security»Domain Account Manager**.

Refer to the *Enabling Enhanced Security* section of this document for more information about changes to security.

Citadel Database Compatibility

The DSC Module 7.0 introduced Citadel 5, a new version of the historical database. The DSC Module 8.0 logs to and retrieves from Citadel 5 databases only. You must convert Citadel 3.8 or Citadel 4.x databases to Citadel 5 if you want to use them with this version of the DSC Module.

You must convert Citadel 3.8 or Citadel 4.x databases before you can log new data to an existing database. If you log data from a converted application before converting the database, the DSC Module creates a new, empty Citadel 5 database in the same location as the older Citadel database and logs data to the new database. If you need to merge logged data from a Citadel 3.8 or Citadel 4.x database with data in a Citadel 5 database, you must use the Historical Data Viewer to convert the older Citadel database and then merge it with the new Citadel 5 database. Refer to the *Historical Data Viewer Help*, available by selecting **Help»Help Topics»Historical Data Viewer** in the NI Measurement & Automation Explorer (MAX), for more information about converting Citadel databases.

Databases Associated with a .scf File

Refer to the *Migrating the .scf File* section for information about converting a database associated with a .scf file.



Note NI recommends using the default conversion options. If you must save the converted database to a location other than the default path, you must update static references to the file paths of the database manually.

Databases Not Associated with a .scf File

If you choose not to convert a Citadel database when the DSC Module 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. Complete the following steps to convert a database.

1. In LabVIEW, select **Tools»DSC Module»View Historical Data** to launch the Historical Data Viewer in MAX.
2. In MAX, right-click the **Historical Data** category and select **Create New** from the shortcut menu to display the **Create New** dialog box.
3. Select the appropriate conversion in the **Create New** dialog box.
4. Click the **Next** button.
5. Enter the path to the database you want to convert.
6. Click the **Finish** button.



Note You cannot convert a database remotely. You must convert the database on the computer where the database exists.

Converting Existing VI-Based Servers

This version of the DSC Module introduces the Custom VI Based Server Assistants to create VI-based servers. Refer to the [Configuring I/O Servers](#) section of this document for more information about the custom VI-based server assistants. Existing VI-based servers continue to work in the DSC Module 8.0 as long as you convert the VI-based server first.

You can convert an existing VI-based server to a VI-based server you can use in the DSC Module 8.0. Complete the following steps to convert a VI-based server.

1. Create a backup of the server VIs.
2. Mass compile the VIs in LabVIEW 8.0.
3. Run the SVRG Add Server Row VI, available in the `vi.lib\lvdsc\legacy\svrlibs\registration` directory, to register the VI-based server. If the server registers correctly, LabVIEW creates a `.xml` file in the `National Instruments\Shared\DSC\VI Based Server\8.0` directory. The `.xml` file is a registration file that specifies that the VI-based server exists.
4. Run the SVR_PackagingWizard VI to build a distribution component of the VI-based server. This VI is available in the `vi.lib\lvdsc\legacy\svrlibs\package`.

If you have an existing VI-based server that does not use the SRVR Get Item Changes VI, you can run only one process at a time with an I/O group connected to the server. If you have different I/O groups from different processes communicating with the server, the server must support dynamic item changes reported by the SRVR Get Item Changes VI. If you need a unique server for each process, create a server using one of the custom VI - based server assistants. VI-based servers share one data space across multiple processes.

Updating OPC Core Components

The DSC Module OPC Client I/O server does not find some OPC 2.x servers when you try to access them over the network. To allow the OPC Client I/O server to locate the OPC 2.x servers, run the OPC Core Components 2.00 Redistributable 2.20.msi, located in the `\Bin\OPCUpdate` directory on the DSC Module 8.0 installation CD, on the computer that is running the OPC 2.x server.

Known Issues

Refer to `readme_DSC.html`, available by selecting **Start»All Programs»National Instruments»LabVIEW 8.2»Readme** or on the LabVIEW DSC Module installation CD, for information about known issues with the DSC Module.

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. 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. In regards to components that use Citadel 5, the following copyright applies. Copyright © 1994 Hewlett-Packard Company. Refer to the *LabVIEW Help* for a listing of the conditions and disclaimers.