

Getting Started with the LabVIEW™ Datalogging and Supervisory Control Module

This document provides an introduction to the LabVIEW Datalogging and Supervisory Control (DSC) Module and contains exercises to help you understand the functionality that the DSC Module provides. To use this document effectively, you should be familiar with LabVIEW, the shared variable, and with building an application in LabVIEW. Refer to the [Related Documentation](#) section of this document for resources to get familiar with LabVIEW.

Contents

What Is the DSC Module?.....	1
Building a DSC Module Application	2
Opening and Running an Example	2
Creating a New Project Library	2
Creating a Periodic I/O Server.....	3
Deploying the Periodic I/O Server	4
Creating Shared Variables	5
Enabling Logging for the Project Library	7
Creating the VI	8
Additional Exercises.....	10
Viewing Real-Time Data.....	10
Viewing Alarms Using the NI Distributed System Manager	11
Monitoring Alarms	11
Related Documentation	11

What Is the DSC Module?

The DSC Module extends the LabVIEW graphical development environment with additional functionality for the rapid development of distributed measurement, control, and high-channel-count monitoring applications.

The DSC Module also enhances the LabVIEW shared variable. Use the shared variable to access and pass data among several VIs in a LabVIEW project or across a network. A shared variable can represent a value or an I/O point. With the DSC Module, you can log data automatically; add alarming, scaling, and security to the shared variable; and configure the shared variable programmatically.

The DSC Module provides tools for graphing historical or real-time trends, enhancing the security of front panels, and writing custom I/O servers. You can read or write to OLE for

Process Control (OPC) connections, Modbus programmable logic controllers (PLC), Experimental Physics and Industrial Control System (EPICS) Client and Server I/O servers, or custom I/O servers that you write. The DSC Module provides solutions for supervisory control of a wide variety of distributed systems using graphical LabVIEW programming.

Building a DSC Module Application

In the following exercises, you will build a VI with DSC Module functionality.

Opening and Running an Example

The DSC Module includes several examples you can use to learn about DSC Module features and applications. In this exercise you will look at an example you will use later to create a custom I/O server.

Complete the following steps to run the DSC Tank Simulator example.

1. Select **Help»Find Examples** to display the NI Example Finder.
2. Select **Directory Structure** in the **Browse according to** section. Browse to the `lvpsc\Getting Started\Tank Simulator` folder and open the DSC Tank Simulator VI. This VI simulates a tank filling with water.
3. Run the VI.
4. Move the sliders to change the rate at which the water fills the tank. As you move the sliders, notice that you control how fast the water flows into the tank.
5. Stop and close the VI.

This VI does not use any DSC Module features. In the exercises that follow, you will create a DSC Module application based on this VI.

Creating a New Project Library

In this exercise you will learn how to create a new LabVIEW project and a project library. The project allows you to manage shared variables, project libraries, and VIs in one window. LabVIEW project libraries are collections of VIs, type definitions, shared variables, palette menu files, and other files, including other project libraries.

Complete the following steps to create a LabVIEW project library.

1. Select **Create Project** in the **Getting Started** window or select **File»Create Project**.
2. Select **Blank Project** and click **Finish** to create a new project.
3. Right-click **My Computer** in the **Project Explorer** window and select **New»Library** from the shortcut menu.
4. Select **File»Save All**. The **Name the Project** dialog box appears.
5. Enter `Tank System` in the **File name** text box and click **OK**. The **Name the Library** dialog box appears.
6. Enter `Tank System IO Server` in the **File name** text box and click **OK**.

You now have a project containing a project library. In the following exercise you will use the project library in the project to create a periodic I/O server.

Creating a Periodic I/O Server

An I/O server is an application that communicates with and manages input/output devices such as PLCs, remote input/output devices, remote Shared Variable Engines, and data acquisition (DAQ) plug-in devices. These I/O servers read selected input items and write to them on demand. The DSC Module can connect to any OPC-compliant I/O server and to many third-party device I/O servers. Refer to the *LabVIEW Help* for more information about creating such servers, such as EPICS, Modbus, and OPC I/O servers. You also can create custom I/O servers. In this exercise, you will build a periodic I/O server. The periodic I/O server will run as a service and publish NI Publish-Subscribe Protocol (NI-PSp) data items to the network.

Complete the following steps to add the periodic I/O server to the project.

1. Right-click the **Tank System IO Server.lvlib** project library in the **Project Explorer** window and select **New»I/O Server** from the shortcut menu. The **Create New I/O Server** dialog box appears.
2. Select **Custom VI – Periodic** from the **I/O Server Type** list and click **Continue**. The **Configure Custom VI – Periodic I/O Server** dialog box appears.
3. Click **New** to display the **Select VI** step of the Custom VI-based Server – Periodic Wizard.
4. Complete the following steps to select the VI from the *Opening and Running an Example* exercise to convert into a periodic I/O server.
 - a. Click the browse button and navigate to the `labview\examples\lvdsc\Getting Started\Tank Simulator` folder.
 - b. Select the DSC Tank Simulator VI.
 - c. Click **Next** in the Custom VI-based Server – Periodic Wizard to advance to the **Select Controls and Indicators To Publish** step.
5. Complete the following steps to select the front panel objects you want to publish as shared variables.
 - a. Remove the checkmark from the **stop** checkbox in the **Controls** list. You will publish the remaining controls and indicators.
 - b. Click **Next** to advance to the **Select Method To Stop The Server** page.
6. Complete the following steps to select the mechanism you want to use to stop the VI.
 - a. Select **Stop the following While Loops**.
 - b. Place a checkmark in the **While Loop** checkbox.

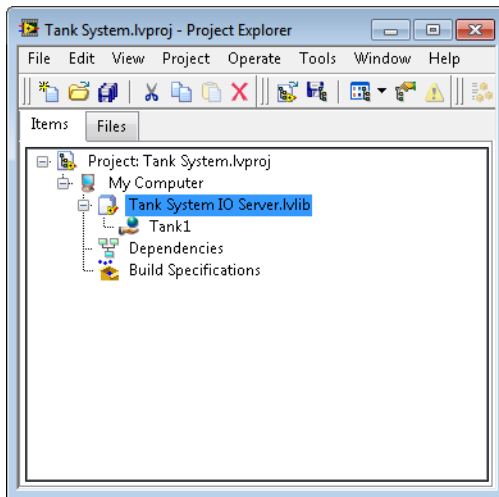


Tip Click **View on block diagram** to highlight the While Loop you select on the block diagram. Use this option if you have multiple While Loops on the block diagram to make sure you select the correct While Loop.

- c. Click **Next** to advance to the **Configure Server Distribution Component** step.

7. You select what you want the wizard to build on the **Configure Server Distribution Component** page. For this exercise, use the default options and click **Next** to advance to the **Server Distribution Component** page.
8. The DSC Module displays a summary of the files that the Custom VI-based Server – Periodic Wizard will create from the **Server Distribution Component** page. Click **Build**. The wizard displays the **Build status** dialog box as it creates a VI template file, a registration VI, and a support DLL and VIs.
9. After the wizard creates the periodic I/O server, the **Configure Custom VI – Periodic I/O Server** dialog box appears with the name of the periodic I/O server and the data items it contains. Click **OK**. LabVIEW adds the periodic I/O server to the Tank System IO Server project library.
10. Expand the **Tank System IO Server.lvlib** project library in the **Project Explorer** window.
11. Right-click the **Custom VI – Periodic1** item and select **Rename** from the shortcut menu. Rename the periodic I/O server **Tank1**. The **Project Explorer** window appears as shown in Figure 1.

Figure 1. Tank System Project



12. Select **File»Save All** in the **Project Explorer** window to save the project, project library, and periodic I/O server.

Deploying the Periodic I/O Server

You must deploy the periodic I/O server so that the data items in the I/O server are available for use in other VIs and across the network. In this exercise you will deploy the periodic I/O server and view the I/O server data in the NI Distributed System Manager.



Note The periodic I/O server runs continuously in the background until you undeploy the library in the **Project Explorer** window that contains the I/O server.

Complete the following steps to deploy the Tank1 periodic I/O server and view the data.

1. Right-click the **Tank System IO Server.lvlib** project library and select **Deploy All** from the shortcut menu to deploy the project library.
2. Click **Close** to close the **Deployment Progress** dialog box when the deployment is complete.
3. Select **Tools»Distributed System Manager**. The NI Distributed System Manager appears. Notice that **Tank System IO Server** appears in the left pane under **My Systems»localhost**.
4. Expand **Tank System IO Server»Tank1**. Notice that the controls and indicators of the I/O server appear under **Tank1**. Because you deployed the project library, the I/O server is running and each control and indicator is an I/O data item.
5. Click **Input Flowrate [GPM]** to view details of the data item in the **Auto View** in the right pane.
6. In the **New Value** text box, enter a value of 10 and click **Set**. Notice that the values of **Tank Level [Gallons]** and **Tank Output Flowrate [GPM]** begin increasing.
7. Close the NI Distributed System Manager. The periodic I/O server continues to run.

Creating Shared Variables

In this exercise you will add the network-published shared variables that represent the data items in the periodic I/O server to the Tank System Shared Variables project library.

Complete the following steps to add the Tank System Shared Variables project library to the Tank System project.

1. Right-click **My Computer** in the **Project Explorer** window and select **New»Library** from the shortcut menu.
2. Right-click the new project library you created and select **Create Bound Variables** from the shortcut menu to display the **Create Bound Variables** dialog box.
3. In **Browse Source**, select **Network Items**.
4. Expand **localhost»Tank System IO Server»Tank1** in the **Network Items** tree. The shared variables appear under **Tank1**.
5. Select each shared variable with the data type **DBL** and click **Add** to add each variable to the **Added variables** list.
6. Click **OK**. The **Create Bound Variables** dialog box closes and the shared variables appear in the **Multiple Variable Editor** window. The **Multiple Variable Editor** window enables you to configure a large number of shared variables at one time.



Note You can configure a single shared variable by using the **Shared Variable Properties** dialog box. Right-click an existing shared variable project item in the **Project Explorer** window and select **Properties** from the shortcut menu to display the **Shared Variable Properties** dialog box.

7. Click **Done** to close the **Multiple Variable Editor** window.
8. Select **File»Save All** and name the project library `Tank System Shared Variables`. LabVIEW binds the shared variables in the `Tank System Shared Variables` project library to the corresponding items on the network.

Configuring Data Logging

When you enable logging for a shared variable, the DSC Module logs shared variable data, including the shared variable value, timestamp, whether the value is in an alarm state, and the quality of the value. The DSC Module can log data to the Citadel database or a supported relational database. In this exercise you will enable logging to the Citadel database for the `Tank Level [Gallons]` shared variable.

Complete the following steps to enable logging for a shared variable.

1. Right-click the **Tank System Shared Variables.lvlib** project library and select **Multiple Variable Editor** from the shortcut menu to open the **Multiple Variable Editor** window.
2. Place a checkmark in the **Logging:Enable** cell for the **Tank Level [Gallons]** shared variable. Additional logging options appear as columns in the table. Make sure the **Logging:Alarms and Events** and **Logging:Data** cells contain checkmarks.
The **Logging:Alarms and Events** option enables alarm and event logging for the shared variable. The **Logging:Data** option enables historical data logging for the shared variable.
3. Enter `0.1` in the **Logging:Resolution** cell for the **Tank Level [Gallons]** shared variable. This value compresses the value to one decimal point when the DSC Module logs the values of this shared variable to the Citadel database.
4. Leave the **Logging:Deadband** cell for the **Tank Level [Gallons]** shared variable with the default value of `0.01`. This value specifies that the DSC Module logs the shared variable value only when the shared variable value differs from the previous shared variable value by 1% of the shared variable engineering scale range, which is 0–100 by default.

Configuring Alarming

An alarm is an abnormal condition on a shared variable or a user-defined condition. An alarm occurs if a shared variable value goes out of its defined alarm limits or if a shared variable has bad status. In this exercise you will add an alarm for the `Tank Level [Gallons]` shared variable.

Complete the following steps to add alarming.

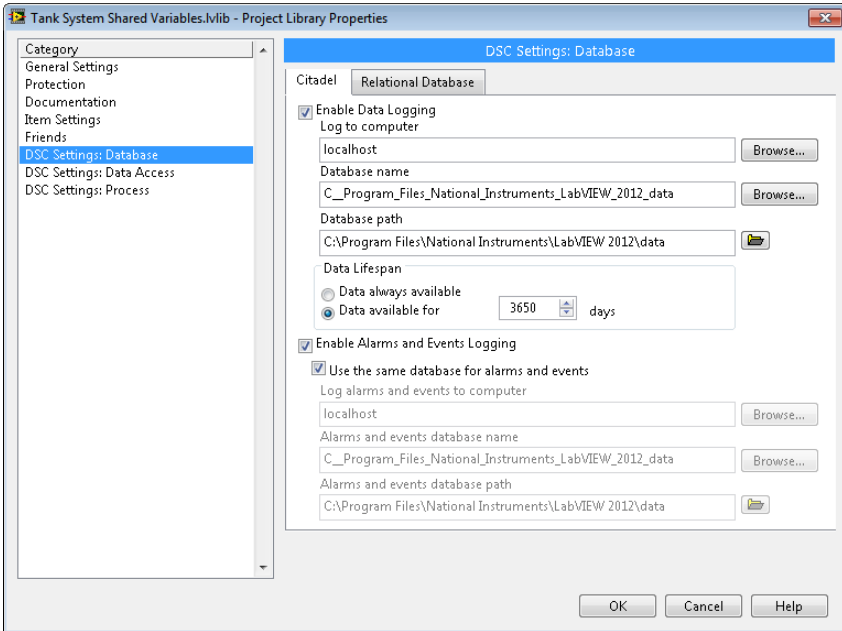
1. Place a checkmark in the **Alarming:Enable** cell for the **Tank Level [Gallons]** shared variable. Additional alarming options appear as columns in the table.
2. Place a checkmark in the **Alarming:HI:Enable** cell for the **Tank Level [Gallons]** shared variable. Additional options for the HI alarm appear as columns in the table. Notice that the default value of the HI alarm is `75`, as shown in the **Alarming:HI:Limit** cell.
3. Click **Done** to apply the changes to the shared variable and to close the **Multiple Variable Editor** window.

Enabling Logging for the Project Library

Complete the following steps to enable data logging and alarm and event logging for the Tank System Shared Variables project library.

1. Right-click the **Tank System Shared Variables.lvlib** project library in the **Project Explorer** window and select **Properties** from the shortcut menu. The **Project Library Properties** dialog box appears.
2. Select **DSC Settings: Database** from the **Category** list.
3. Verify that the options in the **DSC Settings: Database** page appear similar to Figure 2.

Figure 2. DSC Settings: Database Page of the Project Library Properties Dialog Box



The **Enable Data Logging** option turns on data logging for the project library. **localhost** specifies the local computer. Use **localhost** instead of the name of the computer to reduce the changes you must make if you move this project to another computer. The **Enable Alarms and Events Logging** option turns on alarms and events logging for the project library. The **Use the same database for alarms and events** option ensures that the DSC Module logs alarms and events for this project library to the same database that it logs data.



Tip You can change the **Database name** that appears in the **Project Library Properties** dialog box to a more useful or descriptive name.

4. Click **OK** to close the **Project Library Properties** dialog box.

5. Right-click the **Tank System Shared Variables.lvlib** project library in the **Project Explorer** window and select **Deploy All** from the shortcut menu to deploy the shared variables.
6. Click **Close** to close the **Deployment Progress** dialog box when the deployment is complete.
7. Select **File»Save All** in the **Project Explorer** window to save the project, project library, and periodic I/O server.

Creating the VI

A Human Machine Interface (HMI) is the interface through which an operator interacts with a system and with the outside environment that the system monitors and controls. In LabVIEW, you use VIs as the HMI for an application. The DSC Module provides controls, indicators, VIs, and functions to help you design HMIs. In this exercise you will design an HMI that displays data items in an I/O server. You can monitor and control the data items from the LabVIEW front panel.

Complete the following steps to create a VI that uses shared variable nodes to display the data items in the periodic I/O server.

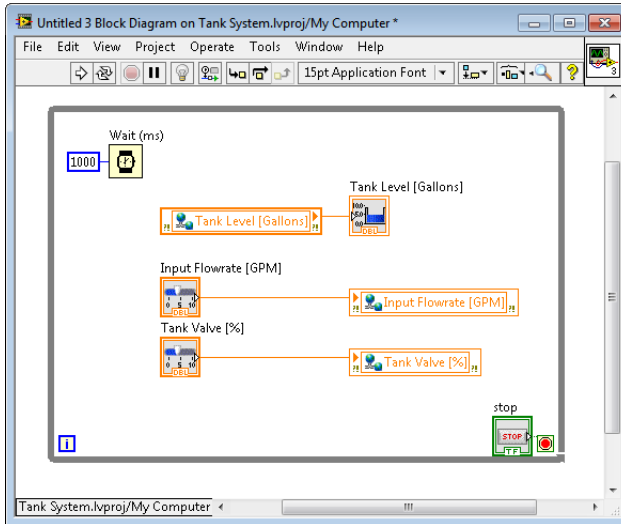
1. Right-click **My Computer** in the **Project Explorer** window and select **New»VI** from the shortcut menu. A new VI front panel and block diagram appear.
2. Select the **Tank Level [Gallons]** shared variable from the **Tank System Shared Variables.lvlib** project library in the **Project Explorer** window and drag the shared variable to the block diagram. The shared variable appears as a shared variable node. By default, LabVIEW configures the shared variable node to read data.
3. Right-click the **Tank Level [Gallons]** output and select **Create»Indicator** from the shortcut menu.
4. Switch to the front panel, right-click the **Tank Level [Gallons]** indicator, and select **Replace»DSC Module»Vessels»Open Tank** from the shortcut menu. You can use the Positioning tool to resize the **Tank Level [Gallons]** tank.
5. Select the **Input Flowrate [GPM]** shared variable from the **Tank System Shared Variables.lvlib** project library in the **Project Explorer** window and drag the shared variable to the block diagram.
6. Right-click the **Input Flowrate [GPM]** shared variable node and select **Access Mode»Write** to configure the shared variable node to write data.
7. Right-click the **Input Flowrate [GPM]** input and select **Create»Control**.
8. Switch to the front panel, right-click the **Input Flowrate [GPM]** control, and select **Replace»Num Ctrls»Pointer Slide**.
9. Repeat steps 5–8 to add the **Tank Valve [%]** shared variable as a control.
10. Place a While Loop around all the objects on the block diagram. The read and write process operations run continuously in the While Loop until an error occurs or until you click the **Stop** button.



Tip Add a Wait function, such as the Wait (ms) function, to help the computer run outside tasks normally without affecting the operation of the loop.

Figure 3 illustrates the block diagram of the VI.

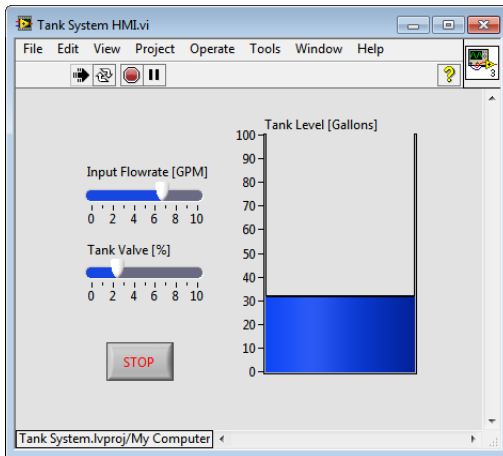
Figure 3. Tank System HMI VI Block Diagram



11. Select **File»Save As** and name the VI **Tank System HMI**.
12. Click **Run** to run the VI.
13. Click **Close** to close the **Deployment Progress** dialog box when the deployment is complete.

Initially, the tank might fill up slowly. Move the sliders to increase or decrease the rate at which the water fills the tank. The VI should function in the same way as the example in the [Opening and Running an Example](#) exercise. Figure 4 illustrates the front panel of the Tank System HMI VI.

Figure 4. Tank System HMI VI Front Panel



14. Click **Stop** to stop the VI.

Additional Exercises

The following exercises introduce some of the additional functionality available with the DSC Module.

Viewing Real-Time Data

You can use the DSC Module to view live data. The Real-Time Trend Express VI displays live data from a shared variable on an XY graph. In the following exercise you will add the ability to view live data to the Tank System HMI VI.

Complete the following steps to add a Real-Time Trend Express VI to a block diagram and view live data.

1. Place the Real-Time Trend Express VI, available on the **DSC Module** palette, on the Tank System HMI VI block diagram. The **Configure Real-Time Trend** dialog box appears.
2. Expand the **Tank System Shared Variables** project library in the **Network Items** tree to display the shared variables.
3. Select the **Tank Level [Gallons]** shared variable and click **Add** in the **Configure Real-Time Trend** dialog box.
4. Click **OK** to close the **Configure Real-Time Trend** dialog box.
5. Add a waveform chart to the front panel.
6. Wire the **Trend Data** output of the Real-Time Trend Express VI to the waveform chart on the block diagram.
7. Place a While Loop around the waveform chart and Real-Time Trend Express VI.

8. Run the VI. The waveform chart shows the water level. Notice that the waveform chart displays the changes when you adjust the **Input Flowrate [GPM]** control or the **Tank Valve [%]** control.

Viewing Alarms Using the NI Distributed System Manager

Complete the following steps to view alarms in the NI Distributed System Manager.

1. Select **Tools»Distributed System Manager** to display the NI Distributed System Manager.
2. Expand the **Tank System Shared Variables** project library in the left pane to display the shared variables.
3. Expand the **Tank Level [Gallons]** shared variable to view alarming information. Notice that the value of **Tank Level [Gallons]»Alarms»HI»level** is 75. This value corresponds to the default value you enabled in the [Configuring Alarming](#) section of this document. The **Tank Level [Gallons]»Alarms»HI** folder contains the values of other alarming properties you enabled.

Monitoring Alarms

Complete the following steps to monitor alarms by using the **Alarms and Events** view in the NI Distributed System Manager.

1. Select **View»Alarms and Events** to display the **Alarms and Events** view.
2. Click **Select Processes** to display the **Select Processes** dialog box.
3. In the **Available Processes** list, expand **My Systems»localhost**.
4. Select **Tank System Shared Variables** and click **Add** to add this process to the **Selected Processes** list.
5. Click **OK** to close the **Select Processes** dialog box.

In this exercise, an alarm appears in the **Alarms and Events** view if the value of **Tank Level [Gallons]** reaches 75.

Related Documentation

Refer to the following documentation for more information about LabVIEW or the DSC Module:

- *Getting Started with LabVIEW*—Use this manual, available by selecting **Help»LabVIEW Help**, as a tutorial to familiarize yourself with the LabVIEW graphical programming environment and the basic LabVIEW features you use to build data acquisition and instrument control applications.
- *LabVIEW Datalogging and Supervisory Control Module Release and Upgrade Notes*—Use these notes, located in the `labview\manuals` directory, to install the DSC Module and learn about new features in the current release.
- *DSC Module Examples*—Refer to the DSC Module examples, located in the `labview\examples\lvds` directory, for example VIs that demonstrate common tasks using the

DSC Module. You also can access these examples in the NI Example Finder by selecting **Help»Find Examples** from LabVIEW and browsing or searching for the DSC Module examples.

- *LabVIEW Help*—Available by selecting **Help»LabVIEW Help** in LabVIEW. Browse the **DSC Module** book on the **Contents** tab for an overview of the DSC Module.
- Visit ni.com/dsc for the latest NI Developer Zone articles, examples, and support information for the DSC Module.
- DSC Module Training—Refer to ni.com/info and enter the Info Code `dsctrn` to access online training for the DSC Module.

LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at ni.com/trademarks for other 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/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patents Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the LabVIEW Datalogging and Supervisory Control Module Readme. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data.

© 2000–2012 National Instruments. All rights reserved.