NI TestStand™ 1: Test Development Exercises

Course Software Version 2012
May 2013 Edition
Part Number 324606R-01

Copyright

End-User License Agreements and Third-Party Legal Notices

Trademarks

Refer to the *NI Trademarks and Logo Guidelines* at `ni.com/trademarks` for more information on National Instruments trademarks.

ARM, Keil, and µVision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group. ©2013 The LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.©2013

FIELDBUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

EtherCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.

CANopen® is a registered Community Trademark of CAN in Automation e.V.

DeviceNet™ and EtherNet/IP™ are trademarks of ODVA.

Go!, SensorDAQ, and Vernier are registered trademarks of Vernier Software & Technology. Vernier Software & Technology and `vernier.com` are trademarks or trade dress.

Xilinx is the registered trademark of Xilinx, Inc.

Taptite and Trilobular are registered trademarks of Research Engineering & Manufacturing Inc.

FireWire® is the registered trademark of Apple Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix®, Tek, and Tektronix, Enabling Technology are registered trademarks of Tektronix, Inc.

The Bluetooth® word mark is a registered trademark owned by the Bluetooth SIG, Inc.

The ExpressCard™ word mark and logos are owned by PCMCIA and any use of such marks by National Instruments is under license.
The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

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

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

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 Patent Notice* at `ni.com/patents`.

Worldwide Technical Support and Product Information

ni.com

Worldwide Offices

Visit 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.

National Instruments Corporate Headquarters

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

To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the Info Code feedback.

Contents

Student Guide

Lesson 1
Introduction to Testing

Lesson 2
Introduction to TestStand

Lesson 3
Analyzing Sequences

Lesson 4
Creating Sequences

Lesson 5
Managing Data

Lesson 6
Overriding Callbacks

Lesson 7
Configuring TestStand

Lesson 8
Executing Tests in Parallel

Lesson 10
User Management

Lesson 11
Deploying a Test System

4

Creating Sequences

Exercises

Exercise 4-1 Creating a Sequence

Goal

Create and run a simple sequence.

Scenario

You are developing a CD test application. The first stage of your development requires you to verify the volume label of the CD. Before you verify the volume label, you must prompt the operator to insert the CD into the drive.

Design

To satisfy the requirements, you must create a new sequence file and insert two steps into the MainSequence. Place a prompt for the operator in the Setup step group and a test for the volume label in the Main step group. Table 4-1 and Table 4-2 describe the properties for each step.

Table 4-1. Insert CD in Drive Prompt Properties

Property	Value
Step Type	Message Popup
Title Expression	`"Insert CD"`
Message Expression	`"Insert the CD under test into the CD drive."`

Table 4-2. Volume Label Test Properties

Property	Value
Step Type	String Value Test
Adapter	LabVIEW
Code Module	`<Exercises>\TestStand 1\CD Test\Code Modules\Volume Information.vi`
Comparison Type	Case Sensitive
Expected String Value	`"500815"`
Parameter: Drive Letter	Directory that contains simulated CD contents, such as `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"`
Parameter: Simulate	`True`
Parameter: Volume Label	`Step.Result.String`
Parameter: error out	`Step.Result.Error`

Note This exercise calls a code module written in LabVIEW. If you are using the LabWindows/CVI development system to complete the course and do not have LabVIEW installed on your computer, refer to the *Student Guide* of the *NI TestStand 1: Test Development* course manual for information about installing and using the LabVIEW Run-Time Engine.

Implementation

1. Create a new sequence file.

   □ Close any open sequence files in the TestStand Sequence Editor.

   □ Select **File»New»Sequence File**.

   □ Save the sequence file as `<Exercises>\TestStand 1\CD Test\CD Test.seq`.

2. Create a Message Popup step to prompt the operator.

□ Expand the Setup step group of the MainSequence.

□ Drag a Message Popup step from the Insertion Palette to the Setup step group of the MainSequence.

Tip Drag the step over the `<Insert Steps Here>` placeholder.

□ Enter `Insert CD in Drive Prompt` as the step name.

□ Click the **Step Settings** pane.

□ Click the **Text and Buttons** tab on the Step Settings pane.

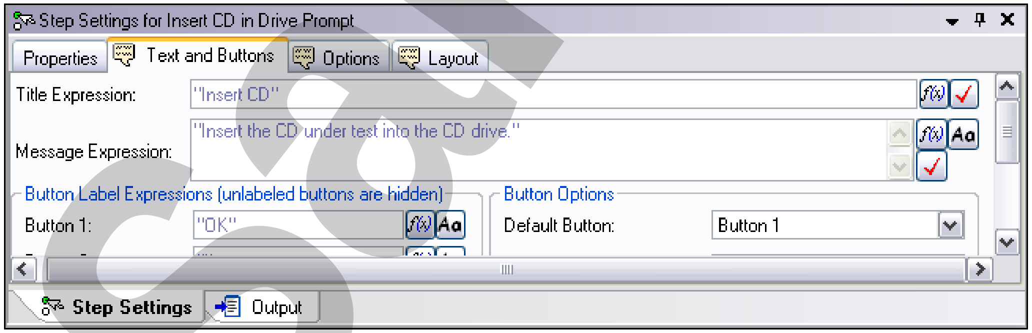
□ Enter `"Insert CD"` in the Title Expression textbox on the Text and Buttons tab.

Note When you are instructed to enter text that includes quotes, you must enter the quotes.

□ Enter `"Insert the CD under test into the CD drive."` in the Message Expression textbox on the Text and Buttons tab.

□ The Step Settings pane should be similar to Figure 4-1.

Figure 4-1. Text and Buttons Settings for Insert CD in Drive Prompt Step



Step Settings for Insert CD in Drive Prompt

Title Expression: "Insert CD"

Message Expression: "Insert the CD under test into the CD drive."

Button Label Expressions (unlabeled buttons are hidden)

Button 1: "OK"

Button Options

Default Button: Button 1

3. Create a String Value Test step to test the CD volume label.

□ Ensure that LabVIEW is selected from the Selected Adapter pull-down menu.

□ Expand the Tests folder on the Insertion Palette.

□ Drag a String Value Test step from the Insertion Palette to the Main step group of the MainSequence.

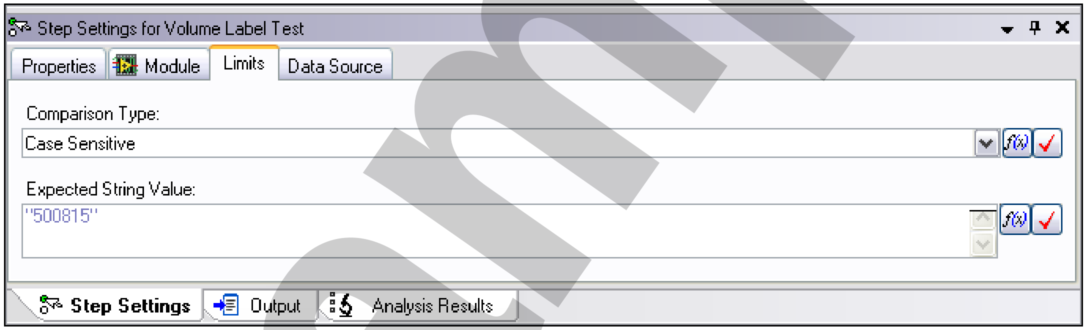
□ Enter `Volume Label Test` as the step name.

□ Select the **Limits** tab on the Step Settings pane.

□ Select **Case Sensitive** from the Comparison Type pull-down menu on the Limits tab.

□ Enter `"500815"` in the Expected String Value textbox on the Limits tab.

Figure 4-2. Limit Settings for Volume Label Test Step



Step Settings for Volume Label Test

Comparison Type:
Case Sensitive

Expected String Value:
"500815"

□ Select the **Module** tab on the Step Settings pane.

□ Next to the VI Path textbox, click **Browse for VI** .

□ Select `<Exercises>\TestStand 1\CD Test\Code Modules\Volume Information.vi`. This module returns a string containing the first six characters of the volume label. This string will be compared to the Expected String Value that you entered on the Step Settings pane.

□ Click **Open**.

- □ In the File Not Found dialog box, select the **Use a relative path for the file you selected** option.

Tip When possible, use relative paths to files so that you can move the project folder to a new location without breaking the file references.

- □ Click **OK**.
- □ In the parameters table on the Module tab, disable the **Default** checkbox for the Simulate parameter.
- □ Enter `True` as the Value for the Simulate parameter.

  The CD Test application must be able to test a real CD and a simulated CD.

  If Simulate is set to True in this code module, this code module simulates reading the volume label from a simulated CD directory the Drive Letter parameter specifies.

  If Simulate is set to False, this code module reads the volume label from an actual CD in the CD drive the Drive Letter parameter specifies.

- □ In the parameters table on the Module tab, disable the **Default** checkbox for the Drive Letter parameter.
- □ Enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Letter parameter. This directory contains the contents of a simulated CD.

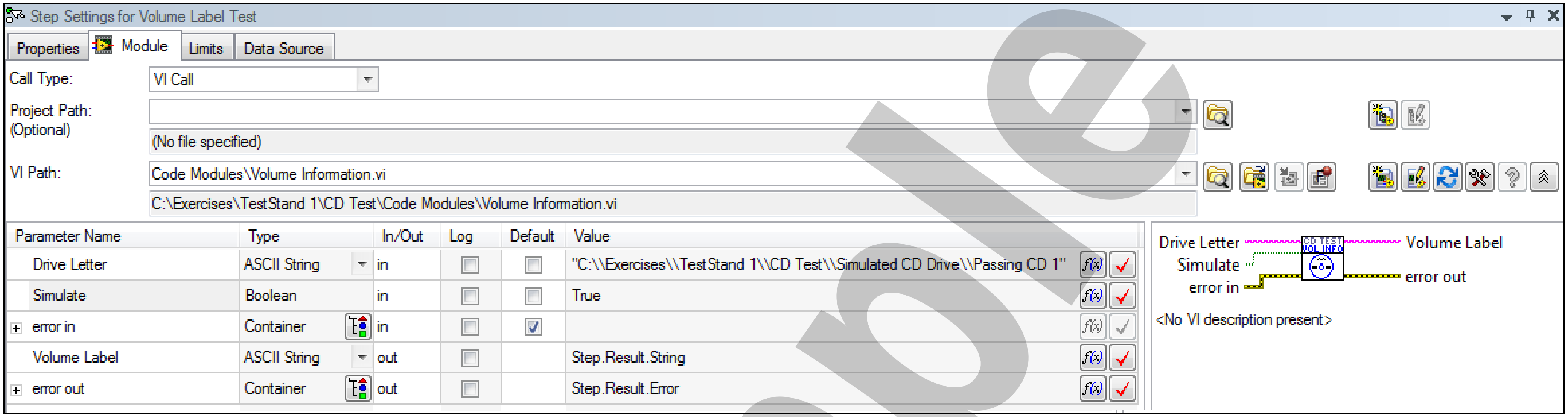
  If files are not installed to the `C:` drive on your computer, change the drive letter to match your computer, such as `D:`. If you change the drive letter, you must do so for the remaining directory paths in this course.

Note You must include two consecutive `\` characters when entering the Drive Letter ASCII string parameter. TestStand uses the single `\` character for special codes inside strings. If you enter a single `\`, TestStand interprets it as a special code and returns an error.

- □ Enter `Step.Result.String` as the Value for the Volume Label parameter.

Note Do not enclose `Step.Result.String` in quotes because it describes data in TestStand rather than a string that TestStand must display or send to the module.

Figure 4-3. Module Settings for Volume Label Test Step



Call Type: VI Call

Project Path: (Optional)

(No file specified)

VI Path: Code Modules\Volume Information.vi

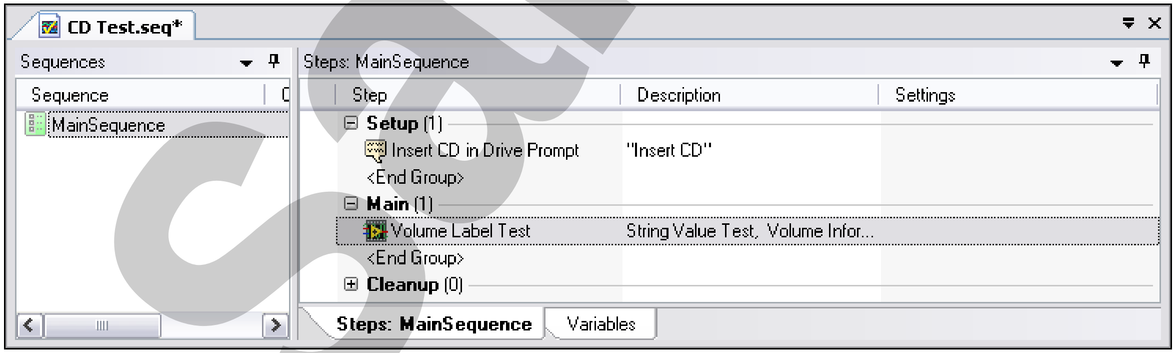
C:\Exercises\TestStand 1\CD Test\Code Modules\Volume Information.vi

Parameter Name	Type	In/Out	Log	Default	Value
Drive Letter	ASCII String	in	□	□	"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"
Simulate	Boolean	in	□	□	True
+ error in	Container	in	□	☑	
Volume Label	ASCII String	out	□		Step.Result.String
+ error out	Container	out	□		Step.Result.Error

<No VI description present>

- □ Verify that `Step.Result.Error` is the Value for the error out parameter.
- □ Verify that your sequence matches Figure 4-4.
- □ Save the sequence file.

Figure 4-4. CD Test Sequence



Steps: MainSequence

Step	Description	Settings
⊟ **Setup** (1)		
Insert CD in Drive Prompt	"Insert CD"	
<End Group>		
⊟ **Main** (1)		
Volume Label Test	String Value Test, Volume Infor...	
<End Group>		
⊞ **Cleanup** (0)		

Testing

1. Test a passing result with a simulated CD that has a volume label of 500815.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, click **OK**.

   □ When the test report appears, verify that the UUT Result is **`Passed`**.

   □ Click the **CD Test.seq** tab to return to the Sequence File window.

2. Test a failing result with a simulated CD that has a volume label of 500111.

   □ Select the Volume Label Test step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Failing CD 1"` as the Value for the Drive Letter parameter. This directory contains the contents of a simulated CD that fails because it contains a volume label not equal to 500815.

   □ Save the sequence file.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, click **OK**.

   □ When the test report appears, verify that the UUT Result is **`Failed`**.

   □ Click the **CD Test.seq** tab to return to the Sequence File window.

3. Test with an error result with a simulated empty CD drive.

   □ Select the Volume Label Test step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\No Disk in Drive"` as the Value for the Drive Letter parameter. This directory simulates the scenario of a CD drive that does not contain a CD.

   □ Save the sequence file.

   □ Select **Execute»Single Pass**.

- □ When the Insert CD dialog box appears, click **OK**.
- □ When the Run-Time Error dialog box appears, review the error code that displays. The current error description is not very informative. You improve the error handling for this sequence later in the course.
- □ Select the **Run Cleanup** option from the Handle Current Error section.
- □ Click **OK**.
- □ Right-click the top of the report tab and select **Close**.
- □ Select the Volume Label Test step.
- □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Letter parameter.
- □ Save the file.

4. (Optional) Test a passing result and an error result with a real CD drive.

Note To complete this optional step, you must use the course CD and the CD drive on a computer running an English-language version of Microsoft Windows 7/Vista/XP.

- □ Select the Volume Label Test step.
- □ In the parameters table on the Module tab, enter `False` as the Value for the Simulate parameter.
- □ In the parameters table on the Module tab, enter the letter of your CD drive as the Value for the Driver Letter parameter. For example, enter `"D:"` if your CD drive is the `D:` drive.
- □ Select **Execute»Single Pass**.
- □ When the Insert CD dialog box appears, load the CD from your course manual into the CD drive.
- □ Click **OK**.
- □ When the test report appears, verify that the UUT Result is **`Passed`**.

□ Click the **CD Test.seq** tab to return to the Sequence File window.

□ Select **Execute»Single Pass**.

□ When the Insert CD dialog box appears, remove all disks from the CD drive.

□ Click **OK**.

□ When the Run-Time Error dialog box appears, review the error code that displays. The current error description is not very informative. You improve the error handling for this sequence later in the course.

□ Select the **Run Cleanup** option from the Handle Current Error section.

□ Click **OK**.

□ Right-click the top of the report tab and select **Close**.

□ Select the Volume Label Test step.

□ In the parameters table on the Module tab, enter `True` as the Value for the Simulate parameter.

□ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Letter parameter.

□ Save the sequence file.

End of Exercise 4-1

Exercise 4-2A Creating a Code Module in LabVIEW

Goal

Create a code module in LabVIEW and call it from TestStand.

Scenario

For the next phase of the CD test, you must analyze the files on the CD. Compare the number of files on the CD to a known number to determine if the correct number of files exists.

Design

Add a Numeric Limit Test step to the current CD Test sequence and use it to compare the result from a code module to a known number of files. Table 4-3 describes the properties for the Numeric Limit Test step.

Table 4-3. File Count Test Properties

Property	Value
Step Type	Numeric Limit Test
Adapter	LabVIEW
Code Module	`<Exercises>\TestStand 1\CD Test\Code Modules\File Count.vi` (created in this exercise)
Comparison Type	EQ
Low Limit	Expected number of files on CD
Units	Files
Parameter: Drive Path	Directory that contains simulated CD contents, such as `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"`
Parameter: Number of Files	`Step.Result.Numeric`
Parameter: error out	`Step.Result.Error`

Create a code module in LabVIEW that recursively searches a directory to obtain the number of files in the directory and all of its subdirectories. The Recursive File List VI performs the search. Table 4-4 describes the inputs and outputs of the code module.

Table 4-4. File Count VI terminals

Type	Name	Properties
Path Control	Drive Path	—
Numeric Indicator	Number of Files	I32 Representation
error out Indicator	error out	—

Implementation

1. Create a Numeric Limit Test step to test the number of files on the CD.

□ Open `<Exercises>\TestStand 1\CD Test\CD Test.seq` if it is not already open.

Note You created `CD Test.seq` in Exercise 4-1. If you have not completed this exercise, open the solution in `<Solutions>\TestStand 1\4-1` directory.

□ Verify that LabVIEW is selected from the Selected Adapter pull-down menu.

□ Expand the Tests folder on the Insertion Palette.

□ Drag a Numeric Limit Test step from the Insertion Palette and insert it below the Volume Label Test step.

□ Enter `File Count Test` for the step name.

□ Click the **Limits** tab on the Step Settings pane.

□ Select **EQ(==)** from the Comparison Type pull-down menu.

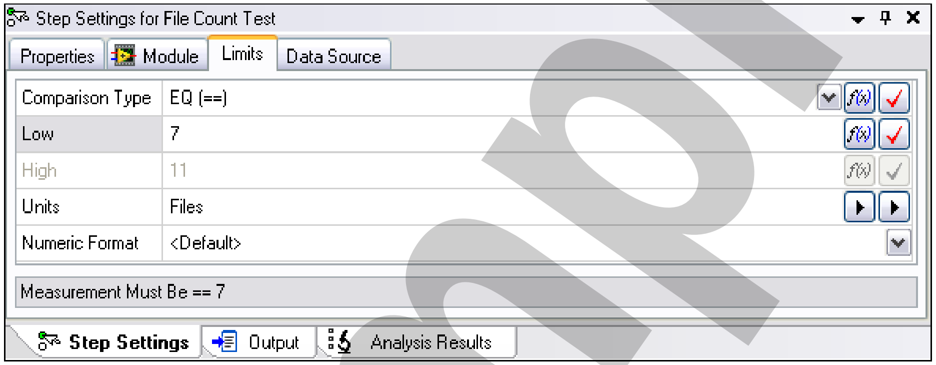
□ Right-click the Windows **Start** button and select **Open Windows Explorer** or **Explore** to launch Windows Explorer.

□ Browse to the `<Exercises>\TestStand 1\CD Test\Simulated CD Drive` directory.

□ Right-click the `Passing CD 1` folder and select **Properties**.

- □ Note the number of files the folder contains in the Properties dialog box.
- □ Click **OK** to exit the Properties dialog box and return to TestStand.
- □ On the Limits tab of the Step Settings pane, enter the number of files in the Passing CD 1 folder for the Low comparison value.
- □ Enter `Files` for the Units.

Figure 4-5. Limit Settings for File Count Test Step



Comparison Type	EQ (==)
Low	7
High	11
Units	Files
Numeric Format	<Default>

Measurement Must Be == 7

2. Create a code module in LabVIEW.
   - □ Click the **Module** tab on the Step Settings pane.
   - □ Click the **Create VI** button .
   - □ Browse to the `<Exercises>\TestStand 1\CD Test\Code Modules\` directory.
   - □ Enter `File Count.vi` for the File Name.
   - □ Click **OK**.
   - □ In the File Not Found dialog box, select the **Use a relative path for the file you selected** option.
   - □ Click **OK**.
3. Configure the terminals of the code module.

□ In LabVIEW, add a File Path Control to the front panel of the File Count VI.

Tip To locate an object in LabVIEW, right-click to display the **Controls** or **Functions** palette, select **Search**, and enter the name of the object you want.

□ Label the control `Drive Path`.

□ Triple-click the label of the Numeric Measurement indicator and change the label to `Number of Files`.

□ Right-click the Number of Files indicator and select **Representation»I32**.

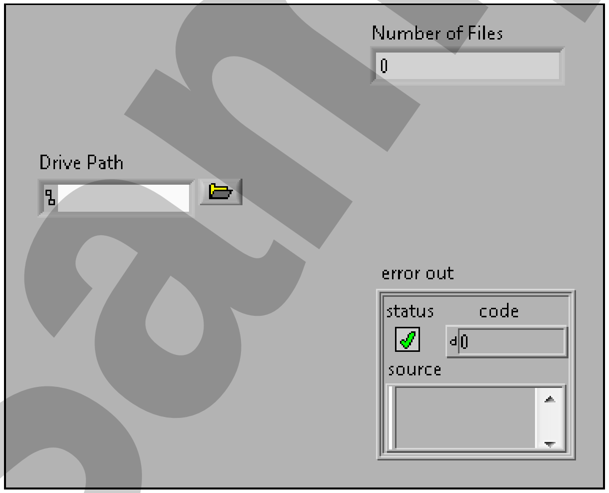
□ Delete the Report Text indicator.

□ In the upper-right corner of the front panel, select the top left terminal of the connector pane.

□ Select the Drive Path control to assign the control to the top left terminal of the connector pane.

□ The front panel should be similar to Figure 4-6.

Figure 4-6. File Count VI Front Panel

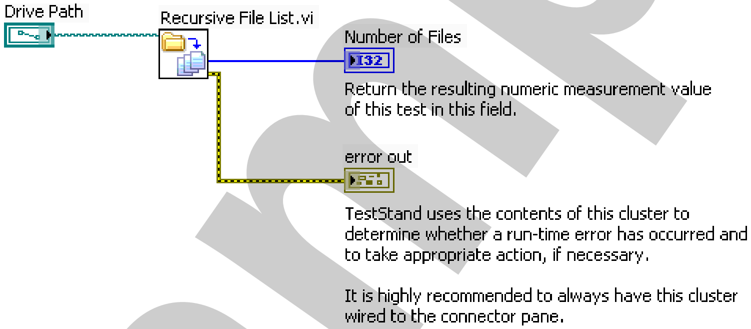


□ Save the VI.

4. Create the block diagram of the code module.

- □ Select **Window»Show Block Diagram or <Ctrl-E>**.
- □ Review the comments on the block diagram, then delete the NOTE comment and the comment that refers to the Report Text terminal you deleted.
- □ Add a Recursive File List VI to the block diagram. To locate this VI, use the palette search feature on the Functions palette.
- □ Wire the **Drive Path** control to the **Folder Path** input of the Recursive File List VI.
- □ Wire the **Number of Files** output of the Recursive File List VI to the **Number of Files** indicator.
- □ Wire the **error out** output of the Recursive File List VI to the **error out** indicator.
- □ Your block diagram should match Figure 4-7.

Figure 4-7. File Count VI Block Diagram



Return the resulting numeric measurement value of this test in this field.

TestStand uses the contents of this cluster to determine whether a run-time error has occurred and to take appropriate action, if necessary.

It is highly recommended to always have this cluster wired to the connector pane.

- □ Save and close the VI.

5. Configure the step settings for the new code module.

- □ Return to TestStand.
- □ Click the **Reload VI Prototype** button on the Module tab.
- □ Click the **Clear all Unmapped Arguments** button in the parameters table of the Module tab.
- □ Disable the **Default** checkbox for the Drive Path parameter.

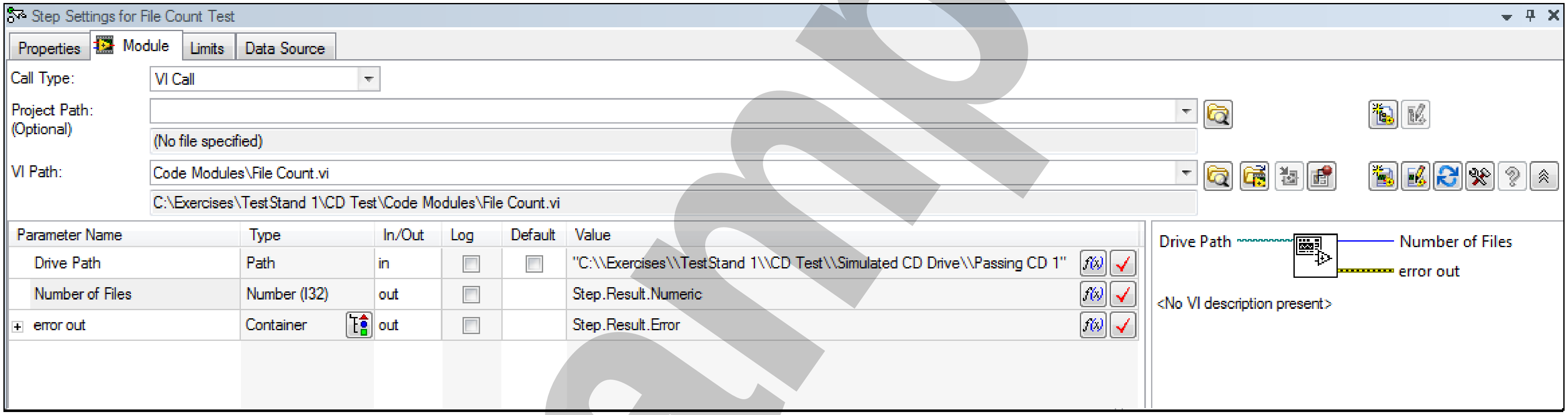
□ Enter the path of a simulated passing CD as the Value for the Drive Path parameter. For example, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"`.

□ Enter `Step.Result.Numeric` as the Value for the Number of Files parameter to specify that you want to compare the limit for this step to the Number of Files output.

□ Verify that `Step.Result.Error` is the Value of the error out parameter to specify that the error out terminal indicates whether an error occurred in the code module.

□ Your Module tab should match Figure 4-8.

Figure 4-8. Module Settings for File Count Test Step



Step Settings for File Count Test

Call Type: VI Call

Project Path: (Optional)

(No file specified)

VI Path: Code Modules\File Count.vi

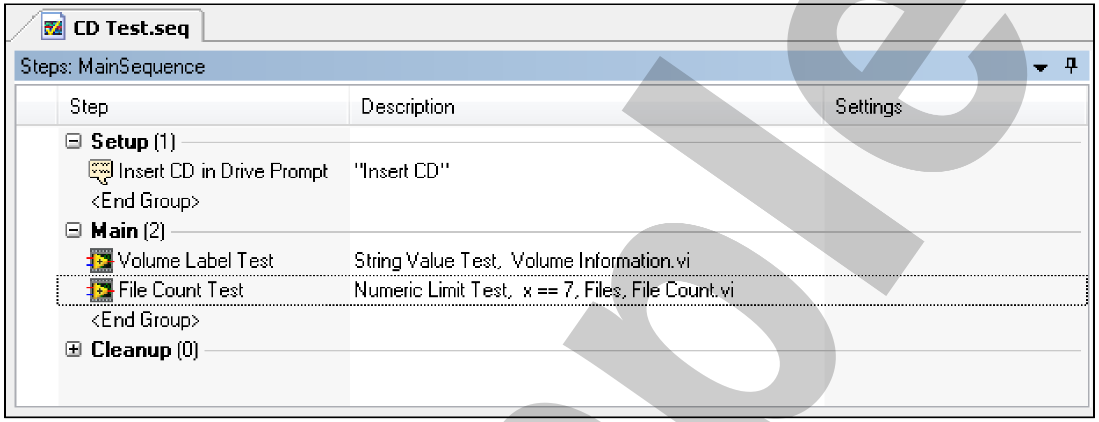
C:\Exercises\TestStand 1\CD Test\Code Modules\File Count.vi

Parameter Name	Type	In/Out	Log	Default	Value
Drive Path	Path	in	□	□	"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"
Number of Files	Number (I32)	out	□		Step.Result.Numeric
+ error out	Container	out	□		Step.Result.Error

<No VI description present>

□ Verify that your sequence matches Figure 4-9.

Figure 4-9. CD Test Sequence



Step	Description	Settings
⊟ **Setup** (1)		
Insert CD in Drive Prompt	"Insert CD"	
<End Group>		
⊟ **Main** (2)		
Volume Label Test	String Value Test, Volume Information.vi	
File Count Test	Numeric Limit Test, x == 7, Files, File Count.vi	
<End Group>		
⊞ **Cleanup** (0)		

□ Save the sequence file.

Testing

1. Test a passing result.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, click **OK**.

   □ Review the test report and locate the Status of the File Count Test to verify that it passed.

   □ Right-click the report tab and select **Close**.

2. (Optional) Test a failing result.

   □ Select the **Volume Label Test** step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Failing CD 1"` as the Value for the Drive Letter parameter.

□ Select the **File Count Test** step.

□ In the parameters table of the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Failing CD 1"` as the Value for the Drive Path parameter. This directory contains the contents of a simulated CD that fails because it does not contain the exact number of files the File Count Test step specifies.

□ Save the sequence file.

□ Select **Execute»Single Pass**.

□ When the Insert CD dialog box appears, click **OK**.

□ Review the test report and locate the Status of the File Count Test to verify that it failed.

□ Right-click the Report tab and select **Close**.

□ Select the **Volume Label Test** step.

□ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Letter parameter.

□ Select the **File Count Test** step.

□ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Path parameter.

□ Save the sequence file.

End of Exercise 4-2A

Exercise 4-2B Creating a Code Module in LabWindows/CVI

Goal

Create a code module in LabWindows/CVI and call it from TestStand.

Scenario

For the next phase of the CD test, you must analyze the files on the CD. Compare the number of files on the CD to a known number to determine if the correct number of files exists.

Design

Add a Numeric Limit Test step to your current CD Test sequence and use it to compare the result from a code module to a known number of files. Table 4-5 describes the properties for the Numeric Limit Test step.

Table 4-5. File Count Test Properties

Property	Value
Step Type	Numeric Limit Test
Adapter	LabWindows/CVI
Code Module	`<Exercises>\TestStand 1\CD Test\Code Modules\File Count.dll` (created in this exercise)
Comparison Type	EQ
Low Limit	Expected number of files on the CD
Units	Files
Function Name	FileCount
Code Template	NumericLimit template for LabWindows/CVI
Parameter: drivePath	Directory that contains simulated CD contents, such as `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"`
Parameter: NumberOfFiles	`Step.Result.Numeric`

Table 4-5. File Count Test Properties (Continued)

Property	Value
Parameter: errorOccurred	`Step.Result.Error.Occurred`
Parameter: errorCode	`Step.Result.Error.Code`
Parameter: errorMsg	`Step.Result.Error.Msg`

Create a code module in LabWindows/CVI that recursively searches a directory to obtain the number of files in the folder and all of its subdirectories. The `<Exercises>\TestStand 1\CD Test\Code Modules\Support C Code\` directory contains `Recursive File Count.h` and `Recursive File Count.c`, which define `RecursiveFileCount` and `FileErrorsToTSErrors` functions.

RecursiveFileCount

```
int RecursiveFileCount(char *Path, long *NumberOfFiles)
```

The RecursiveFileCount function searches through a directory and all of its subdirectories and returns the cumulative count of files in the directories. This function does not include the directories themselves in the file count, only their contents.

Inputs: **Path**—A string that defines the root path to search. The `RecursiveFileCount` function modifies this parameter for each recursive call, and thus the output value of this parameter is undefined.

Outputs: **NumberOfFiles**—An integer that indicates the total number of files. A return value of `0` indicates success. A negative value indicates that an error has occurred. This function returns the same set of error codes as the `GetFirstFile` and `GetNextFile` Utility Library functions. Use the `FileErrorsToTSErrors` function to convert the return value to TestStand error codes.

FileErrorsToTSErrors

```
long FileErrorsToTSErrors (int result)
```

The `FileErrorsToTSErrors` function converts the error codes the `GetFirstFile`, `GetLastFile`, or `RecursiveFileCount` functions return to the corresponding TestStand error codes.

Inputs: **result**—A code returned by the `GetFirstFile`, `GetLastFile`, or `RecursiveFileCount` functions.

Outputs: **Return value**—A TestStand error code.

The code module must allocate memory, call the functions, and handle the return data. Table 4-6 describes the inputs and outputs of the code module.

Table 4-6. FileCount Function Prototype

Parameter Name	Type	Properties
Return Value	void	—
drivePath	C String (const)	—
numberOfFiles	Signed 32-bit Integer	Pass by Reference
errorOccured	Signed 16-bit Integer	Pass by Reference
errorCode	Signed 32-bit Integer	Pass by Reference
errorMsg	C String Buffer	Buffer size 1024

Implementation

1. Create a Numeric Limit Test step to test the number of files on the CD.

□ Open `<Exercises>\TestStand 1\CD Test\CD Test.seq` if it is not already open.

Note You created the `CD Test.seq` in Exercise 4-1. If you have not completed this exercise, open the solution in `<Solutions>\TestStand 1\4-1`.

□ Select **LabWindows/CVI** from the Selected Adapter pull-down menu.

□ Expand the **Tests** folder on the Insertion Palette.

□ Drag a Numeric Limit Test step from the Insertion Palette and insert it below the Volume Label Test step on the Steps pane.

□ Enter `File Count Test` for the step name.

□ Click the **Limits** tab on the Step Settings pane.

□ Select **EQ(==)** from the Comparison Type pull-down menu.

□ Right-click the Windows **Start** button and select **Open Windows Explorer** or **Explore** to launch Windows Explorer.

□ Browse to the `<Exercises>\TestStand 1\CD Test\Simulated CD Drive` directory.

□ Right-click the `Passing CD 1` folder and select **Properties**.

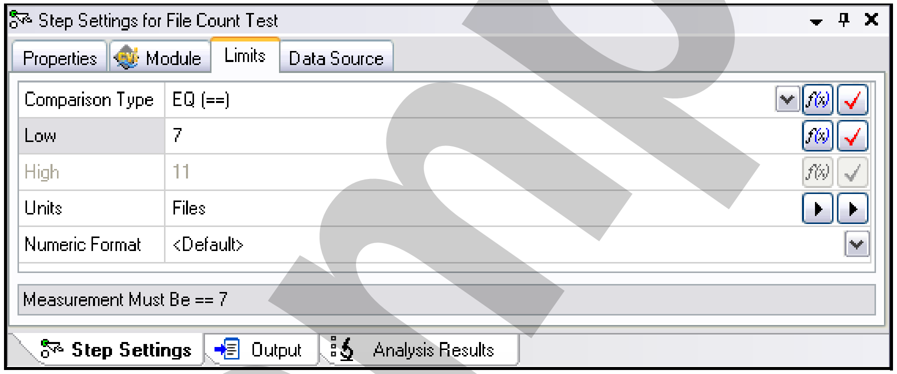
□ Note the number of files the folder contains in the Properties dialog box.

□ Click **OK** to exit the Properties dialog box and return to TestStand.

□ On the Limits tab of the Step Settings pane, enter the number of files in the Passing CD 1 folder for the Low comparison value.

□ Enter `Files` as the Units, as shown in Figure 4-10.

Figure 4-10. Limit Settings for File Count Test Step



Comparison Type	EQ (==)
Low	7
High	11
Units	Files
Numeric Format	<Default>

Measurement Must Be == 7

2. Define the prototype of a LabWindows/CVI code module.

□ Click the **Module** tab on the Step Settings pane.

□ Enter `FileCount` as the Function.

□ Click the **Add Parameter** button + .

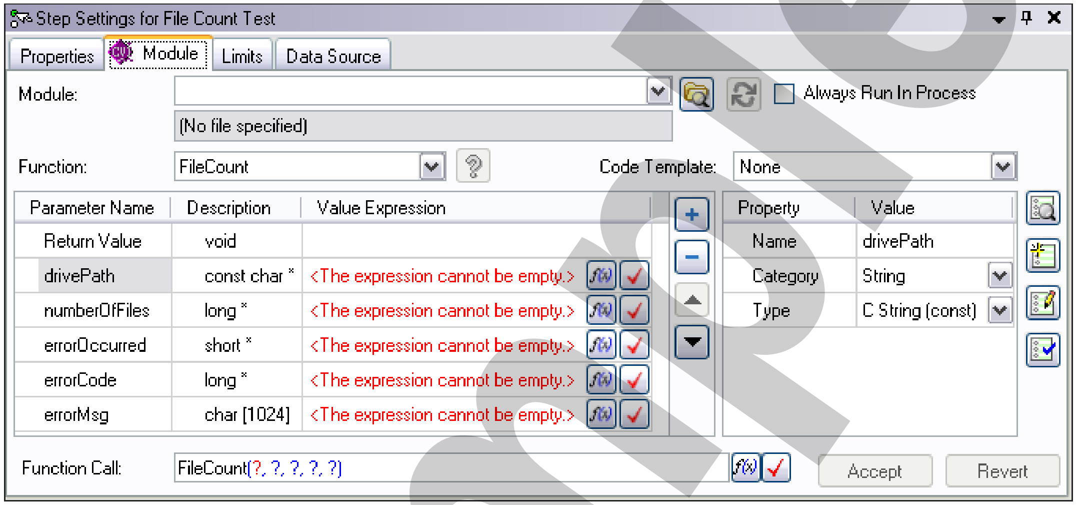
□ Enter `drivePath` for the Name property.

□ Select **String** for the Category property.

- □ Select **C String (const)** for the Type property.
- □ Click **Add Parameter**.
- □ Enter `numberOfFiles` for the Name property.
- □ Select **Numeric** for the Category property.
- □ Select **Signed 32-bit Integer** for the Type property.
- □ Select **By Reference (by pointer)** for the Pass property.
- □ Click **Add Parameter**.
- □ Enter `errorOccurred` for the Name property.
- □ Select **Numeric** for the Category property.
- □ Select **Signed 16-bit Integer** for the Type property.
- □ Select **By Reference (by pointer)** for the Pass property.
- □ Click **Add Parameter**.
- □ Enter `errorCode` for the Name property.
- □ Select **Numeric** for the Category property.
- □ Select **Signed 32-bit Integer** for the Type property.
- □ Select **By Reference (by pointer)** for the Pass property.
- □ Click **Add Parameter**.
- □ Enter `errorMsg` for the Name property.
- □ Select **String** for the Category property.
- □ Select **C String Buffer** for the Type property.

- □ Verify that the Buffer Size property is set to `1024`.
- □ Verify that your Module tab matches Figure 4-11.

Figure 4-11. Module Settings for File Count Test Step



Module:

☐ Always Run In Process

[No file specified]

Function: FileCount

Code Template: None

Parameter Name	Description	Value Expression
Return Value	void	
drivePath	const char *	<The expression cannot be empty.>
numberOfFiles	long *	<The expression cannot be empty.>
errorOccurred	short *	<The expression cannot be empty.>
errorCode	long *	<The expression cannot be empty.>
errorMsg	char [1024]	<The expression cannot be empty.>

Property	Value
Name	drivePath
Category	String
Type	C String (const)

Function Call: FileCount(?, ?, ?, ?, ?)

3. Create the LabWindows/CVI code module.

- □ Click the **Create Code** button .
- □ Browse to the `<Exercises>\TestStand 1\CD Test\Code Modules\` directory.
- □ Enter `File Count.prj` as the File Name.
- □ Click **OK**.
- □ In the File Not Found dialog box, select the **Use a relative path for the file you selected** option.
- □ Click **OK**.
- □ Browse to the `<Exercises>\TestStand 1\CD Test\Code Modules\` directory.

□ Enter File Count.c as the File Name.

□ Click **OK**.

□ In the File Not Found dialog box, select the **Use a relative path for the file you selected** option.

□ Click **OK**.

4. Include the necessary source and header files.

□ In LabWindows/CVI, right-click the Include Files folder of the project tree and select **Add Existing File**.

□ Select <Exercises>\TestStand 1\CD Test\Code Modules\Support C Code\Recursive File Count.h.

□ Click **OK**.

□ Right-click the Source Files folder of the project tree and select **Add Existing File**.

□ Select <Exercises>\TestStand 1\CD Test\Code Modules\Support C Code\Recursive File Count.c.

□ Click **OK**.

□ Add the following line to the top of the File Count.c file:

```
#include "Recursive File Count.h"
```

5. Allocate a new copy of the path that the RecursiveFileCount function can modify.

□ Add the following lines to the FileCount function below the //Insert function body here comment.

```
char modifiedPath[MAX_PATHNAME_LEN];
strcpy (modifiedPath, drivePath);
```

6. Initialize the numberOfFiles parameter to zero.

□ Add the following line to the FileCount function.

```
*numberOfFiles = 0;
```

7. Call the `RecursiveFileCount` function and convert its return value to an error code that TestStand can recognize.

   □ Add the following line to the `FileCount` function.

```
*errorCode = FileErrorsToTSErrors(RecursiveFileCount(modifiedPath,numberOfFiles));
```

Note The `Recursive File Count.c` file defines the `FileErrorsToTSErrors` function and the `RecursiveFileCount` function. Open the source file to review the implementation of the functions.

8. Process any errors.

   □ Add the following lines to the `FileCount` function.

```
if(*errorCode < 0)
{
   *errorOccurred = TRUE;
   TS_ErrorCodeToString(*errorCode, 0, TRUE, &errorMsg);
}
```

9. Compile the code into a dynamically linked library for TestStand to call.

   □ Confirm that your code matches the following code. The code that you modified is shown in bold.

```
#include "Recursive File Count.h"
void __declspec(dllexport) FileCount(const char *drivePath, long *numberOfFiles, short *errorOccurred, long *errorCode,
char errorMsg[1024])
{
   char modifiedPath[MAX_PATHNAME_LEN];
   strcpy (modifiedPath, drivePath);
   *numberOfFiles = 0;
   *errorCode = FileErrorsToTSErrors(RecursiveFileCount(modifiedPath, numberOfFiles));
   if(*errorCode < 0)
   {
      *errorOccurred = TRUE;
      TS_ErrorCodeToString(*errorCode, 0, TRUE, &errorMsg);
   }
}
```

   □ Select **File»Save All**.

□ Select **Build»Create Debuggable Dynamic Link Library**.

□ If prompted to add `#include` directives for `ansi_c.h` and `tsutil.h`, click **Yes**.

□ If any errors occur, check your code against the code shown in this step, correct the errors, and rebuild the debuggable dynamic link library.

□ When you are informed that the `File Count.dll` and other files have been created, click **OK**.

10. Configure the step settings for the new code module.

□ Return to TestStand.

□ Click the **Browse** button on the Module tab.

□ Browse to the `<Exercises>\TestStand 1\CD Test\Code Modules\` directory.

□ Select `File Count.dll`.

□ Click **Open**.

□ In the File Not Found dialog box, select the **Use a relative path for the file you selected** option.

Tip You might need to expand the size of the Step Settings pane or scroll through the parameter list to see all of the parameters.

□ Enter the path of a simulated passing CD as the Value for the drivePath parameter. For example, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"`.

□ Enter `Step.Result.Numeric` as the Value Expression for the numberOfFiles parameter to specify that you want to compare the limit for this step to the Number of Files output.

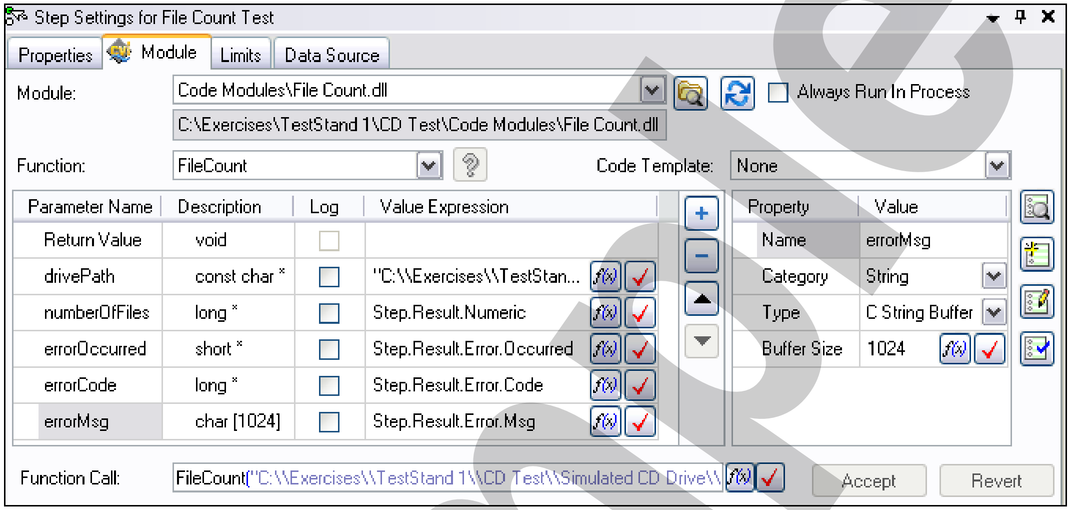
□ Enter `Step.Result.Error.Occurred` as the Value Expression for the errorOccurred parameter to check this parameter to determine if an error occurred in the code mode.

□ Enter `Step.Result.Error.Code` as the Value Expression for the errorCode parameter. TestStand uses this code to identify the error if `Step.Result.Error.Occured` is `True`.

□ Enter `Step.Result.Error.Msg` as the Value Expression for the errorMsg parameter. TestStand uses this string to report an error message if `Step.Result.Error.Occured` is `True`.

□ Your Module tab should match Figure 4-12.

Figure 4-12. Module Settings for File Count Test Step



Module: Code Modules\File Count.dll

C:\Exercises\TestStand 1\CD Test\Code Modules\File Count.dll

□ Always Run In Process

Function: FileCount

Code Template: None

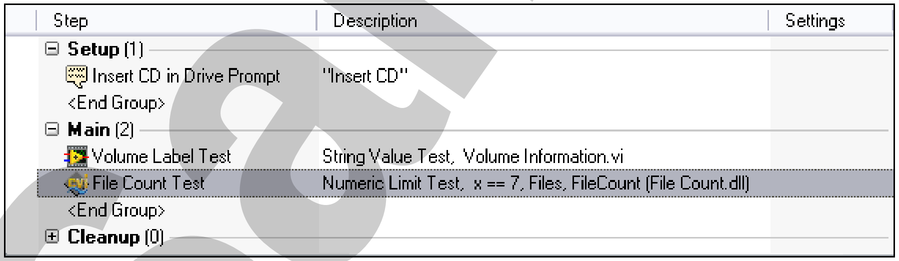
Parameter Name	Description	Log	Value Expression
Return Value	void	□	
drivePath	const char *	□	"C:\\Exercises\\TestStan...
numberOfFiles	long *	□	Step.Result.Numeric
errorOccurred	short *	□	Step.Result.Error.Occurred
errorCode	long *	□	Step.Result.Error.Code
errorMsg	char [1024]	□	Step.Result.Error.Msg

Property	Value
Name	errorMsg
Category	String
Type	C String Buffer
Buffer Size	1024

Function Call: FileCount("C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\

□ Your sequence should match Figure 4-13.

Figure 4-13. CD Test Sequence



Step	Description	Settings
⊟ **Setup** (1)		
Insert CD in Drive Prompt	"Insert CD"	
<End Group>		
⊟ **Main** (2)		
Volume Label Test	String Value Test, Volume Information.vi	
File Count Test	Numeric Limit Test, x == 7, Files, FileCount (File Count.dll)	
<End Group>		
⊞ **Cleanup** (0)		

□ Save the sequence file.

Testing

1. Test a passing result.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, click **OK**.

   □ Review the test report and locate the Status of the File Count Test to verify that it passed.

   □ Right-click the top of the report tab and select **Close**.

2. (Optional) Test a failing result.

   □ Select the **Volume Label Test** step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Failing CD 1"` as the Value for the Drive Letter parameter.

   □ Select the **File Count Test** step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Failing CD 1"` as the Value for the drivePath parameter. This directory contains the contents of a simulated CD that fails because it does not contain the exact number of files the File Count Test step specifies.

   □ Save the sequence file.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, click **OK**.

   □ Review the test report and locate the Status of the File Count Test to verify that it failed.

   □ Right-click the top of the report tab and select **Close**.

   □ Select the **Volume Label Test** step.

   □ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the Drive Letter parameter.

□ Select the **File Count Test** step.

□ In the parameters table on the Module tab, enter `"C:\\Exercises\\TestStand 1\\CD Test\\Simulated CD Drive\\Passing CD 1"` as the Value for the drivePath parameter.

□ Save the files.

End of Exercise 4-2B

Exercise 4-3 Creating a Subsequence

Goal

Construct a simple subsequence.

Scenario

The next phase of the test uses image processing to test the physical dimensions of the CD and the label printed onto the CD. This portion of the test includes subtests for the CD size, logo, title, version, and the part number. You can place the steps that implement these tests in a subsequence to keep the image recognition tests separate from the rest of your code, make the tests easier to debug, and allow you to call the tests from other sequences.

You develop the image recognition tests later, but first you must construct a subsequence that contains the tests. For this exercise, the subsequence prompts the user to place the CD under a camera but performs no additional actions.

Design

Create a subsequence named CD Image Recognition. Insert a Message Popup step in the sequence to prompt the user to place the CD under the camera. Insert a Label step as a reminder and placeholder for the image recognition steps. Table 4-7 and Table 4-8 describe the properties for each step.

Table 4-7. Place CD Under Camera Prompt Properties

Property	Value
Step Type	Message Popup
Title Expression	`"Move CD"`
Message Expression	`"Place the CD under test in front of the camera."`

Table 4-8. Insert Image Recognition Steps Here Properties

Property	Value
Step Type	Label
Name	`"Insert Image Recognition Steps Here"`

Implementation

1. Create a subsequence to handle image recognition.

   □ Open `<Exercises>\TestStand 1\CD Test\CD Test.seq` if it is not already open.

   Note You last modified the `CD Test.seq` in Exercise 4-2. If you have not completed this exercise, open the solution in `<Solutions>\TestStand 1\4-2`.

   □ Right-click in the Sequences pane and select **Insert Sequence**.

   □ Enter `CD Image Recognition` as the sequence name.

2. Create a Message Popup step to prompt the operator.

   □ Drag a Message Popup step from the Insertion Palette to the Setup step group of the CD Image Recognition sequence.

   □ Enter `Place CD Under Camera Prompt` as the step name.

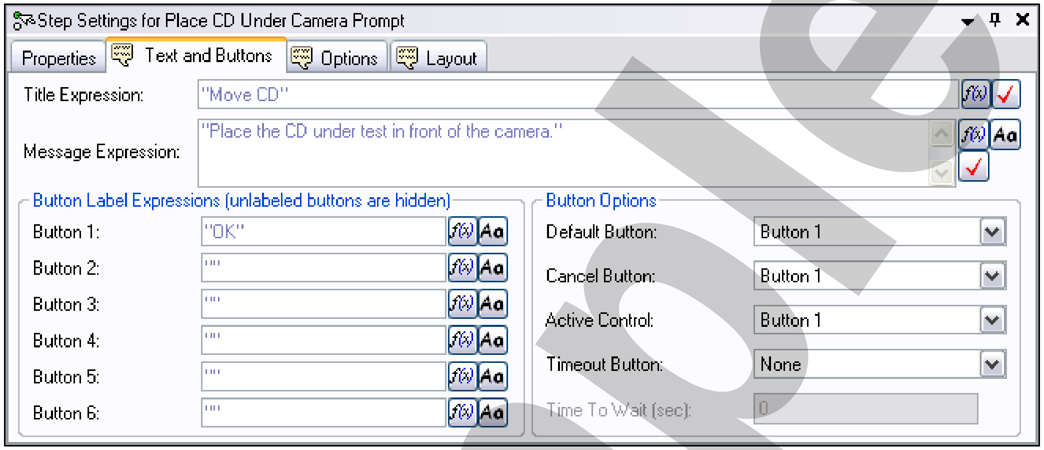
   □ Click the **Text and Buttons** tab on the Step Settings pane.

   □ Enter `"Move CD"` in the Title Expression textbox.

   □ Enter `"Place the CD under test in front of the camera."` in the Message Expression textbox.

- □ Your Text and Buttons tab settings should match Figure 4-16.

Figure 4-14. Text and Buttons Settings for Place CD Under Camera Prompt Step



Step Settings for Place CD Under Camera Prompt

Title Expression: "Move CD"

Message Expression: "Place the CD under test in front of the camera."

Button Label Expressions (unlabeled buttons are hidden)

Button 1:	"OK"
Button 2:	""
Button 3:	""
Button 4:	""
Button 5:	""
Button 6:	""

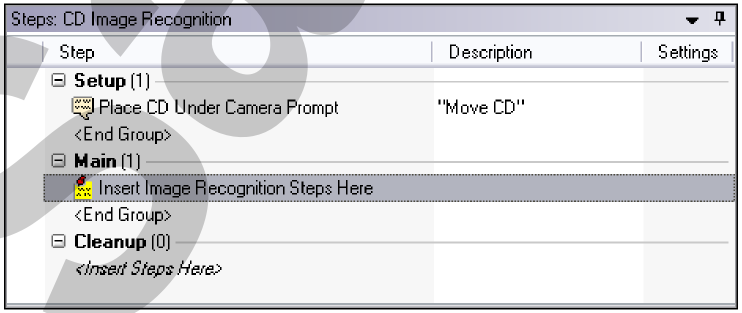
Button Options

Default Button:	Button 1
Cancel Button:	Button 1
Active Control:	Button 1
Timeout Button:	None
Time To Wait (sec):	0

3. Add a Label step as a comment and placeholder.

- □ Drag a Label step from the Insertion Palette to the Main step group of the CD Image Recognition sequence.
- □ Enter `Insert Image Recognition Steps Here` as the step name.
- □ Your subsequence should match Figure 4-15.

Figure 4-15. CD Image Recognition Sequence



Steps: CD Image Recognition

Step	Description	Settings
⊟ **Setup** (1)		
Place CD Under Camera Prompt	"Move CD"	
<End Group>		
⊟ **Main** (1)		
Insert Image Recognition Steps Here		
<End Group>		
⊟ **Cleanup** (0)		
<Insert Steps Here>		

4. Call the CD Image Recognition subsequence from the MainSequence.

   □ Select the MainSequence on the Sequences pane.

   □ Drag a Sequence Call step from the Insertion Palette and insert it below the File Count Test step on the Steps pane.

   □ Enter `Call CD Image Recognition` as the step name.

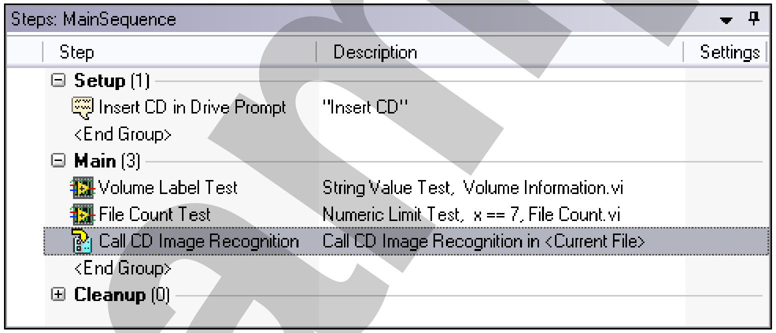
   □ Click the **Module** tab on the Step Settings pane.

   □ Enable the **Use Current File** checkbox.

   □ Select **CD Image Recognition** from the Sequence pull-down menu.

   □ Your sequence should match Figure 4-16. If you completed Exercise 4-2 B, then the File Count Test step displays a LabWindows/CVI icon instead of a LabVIEW icon.

Figure 4-16. CD Test Sequence



Steps: MainSequence

Step	Description	Settings
⊟ **Setup** (1)		
Insert CD in Drive Prompt	"Insert CD"	
<End Group>		
⊟ **Main** (3)		
Volume Label Test	String Value Test, Volume Information.vi	
File Count Test	Numeric Limit Test, x == 7, File Count.vi	
Call CD Image Recognition	Call CD Image Recognition in <Current File>	
<End Group>		
⊞ **Cleanup** (0)		

   □ Save the files.

Testing

1. Test the operator prompt.

   □ Select **Execute»Single Pass**.

   □ When the Insert CD dialog box appears, and click **OK**.

   □ The Move CD dialog box opens to indicate that TestStand called the subsequence.

   □ Click **OK**.

   □ Right-click the report tab and select **Close**.

End of Exercise 4-3