NI TestStand™ I: Introduction Course Manual

Course Software Version 4.0
July 2007 Edition
Part Number 322317J-01

Contents

Lesson 4
Creating Sequences

Lesson 5
Managing Data

Lesson 6
Overriding Callbacks

Lesson 7
Configuring TestStand

Lesson 8
Executing Tests in Parallel

Lesson 9
Best Practices for Test Development

Lesson 10
User Management

Lesson 11
Deploying a Test System

Appendix A
Additional Information and Resources

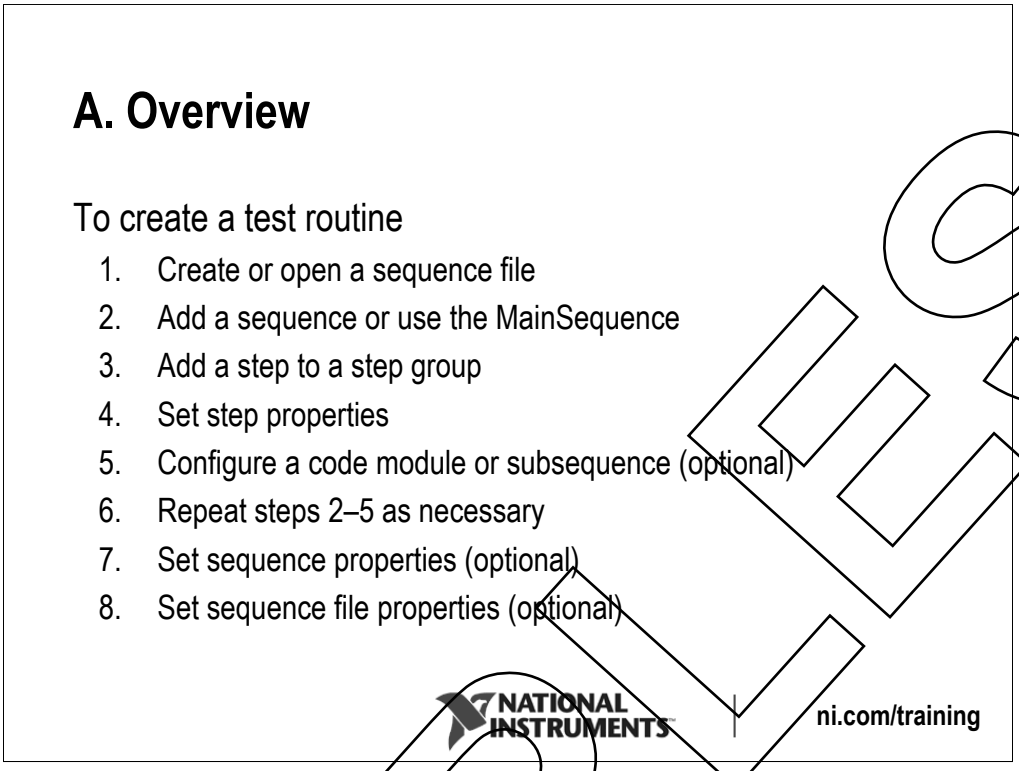
Course Evaluation

Lesson 4: Creating Sequences

TOPICS

A. Overview
B. Creating Steps
C. Code Modules
D. Subsequences
E. Sequence Properties
F. Sequence File Properties

NATIONAL INSTRUMENTS | ni.com/training

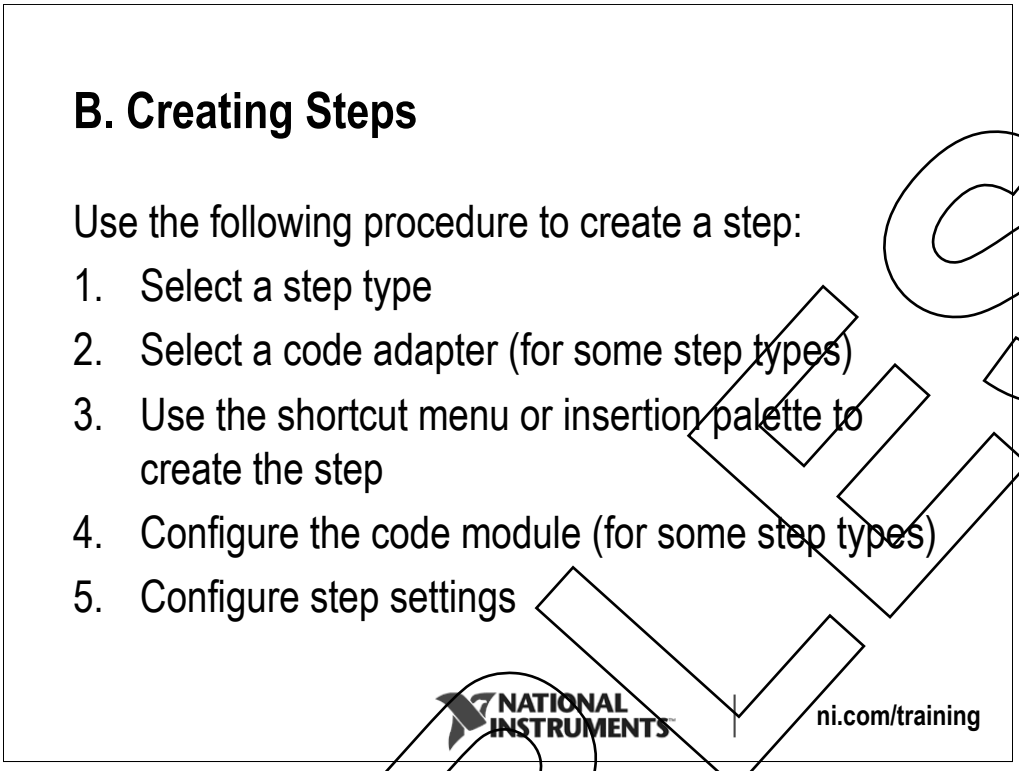


A. Overview

Use the following procedure to create a test routine:

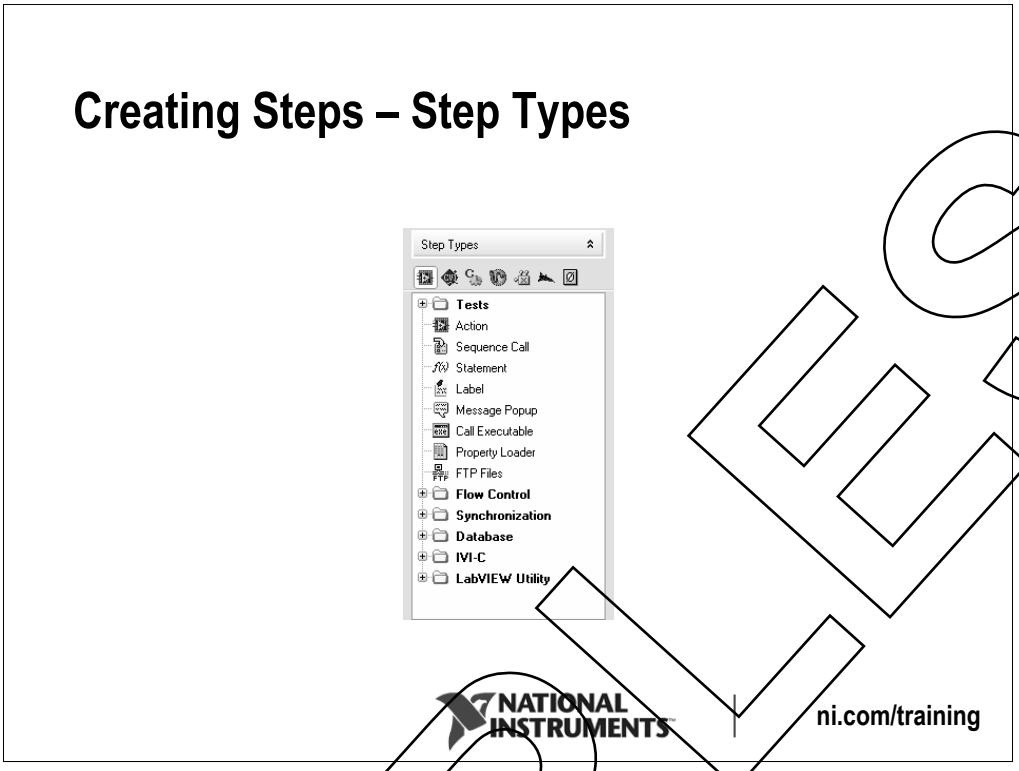
1. Create or open a sequence file—Use the File menu or the toolbar.
2. Add a sequence or use the MainSequence—When you create a new sequence file, TestStand automatically creates a MainSequence. To create other sequences, right-click the Sequences pane and select **Insert Sequence**.
3. Add a step—Drag a step from the Insertion Palette to a step group in the Steps pane or right-click in the Steps pane and select **Insert Step**.
4. Set step properties—All steps have properties that define their behavior. Use the tabs in the Step Settings pane to configure these properties.
5. Configure a code module or subsequence—Some steps call other sequences or code modules written in other languages. For these steps, use the Module tab on the Step Settings pane to specify the location of the module you want to call, the data you want to pass to the module, and the storage location of any data that the module returns.
6. Repeat steps 2–5 as necessary—Repeat the procedure for each step you need to add to your sequence.

7. Set sequence properties—The sequence contains properties that you can access by right-clicking the sequence in the Sequences pane and selecting Sequence Properties. Most sequences use the default properties, but you may want to configure a precondition for the sequence to execute, specify how the sequence reacts when a critical step fails, or enter documentation and requirements information.
8. Set sequence file properties—To set properties of the sequence file as a whole, select **Edit»Sequence File Properties**. Use sequence file properties to specify documentation, memory, file format, and other options. A sequence file option of particular importance is the Model Option property, which you can use to specify a process model for the sequence file to use.



B. Creating Steps

To create a step, select a step type to insert in the sequence, then specify the settings for the step. Step settings include properties of the step and step-specific settings. You can also use step templates to create steps.

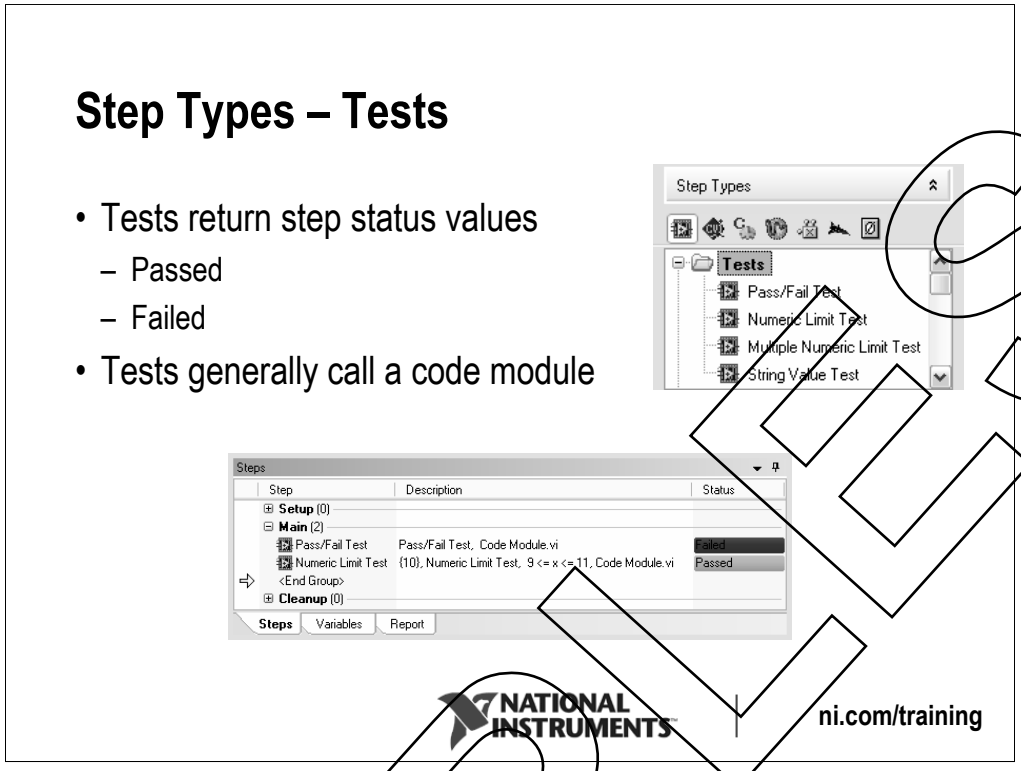


Creating Steps – Step Types

When you insert a step, you must first choose a step type. The step type defines the behavior and configuration options of the step. TestStand includes a wide variety of built-in step types. You can also create and use custom step types. Refer to the *NI TestStand II: Customization* course for more information about custom step types.

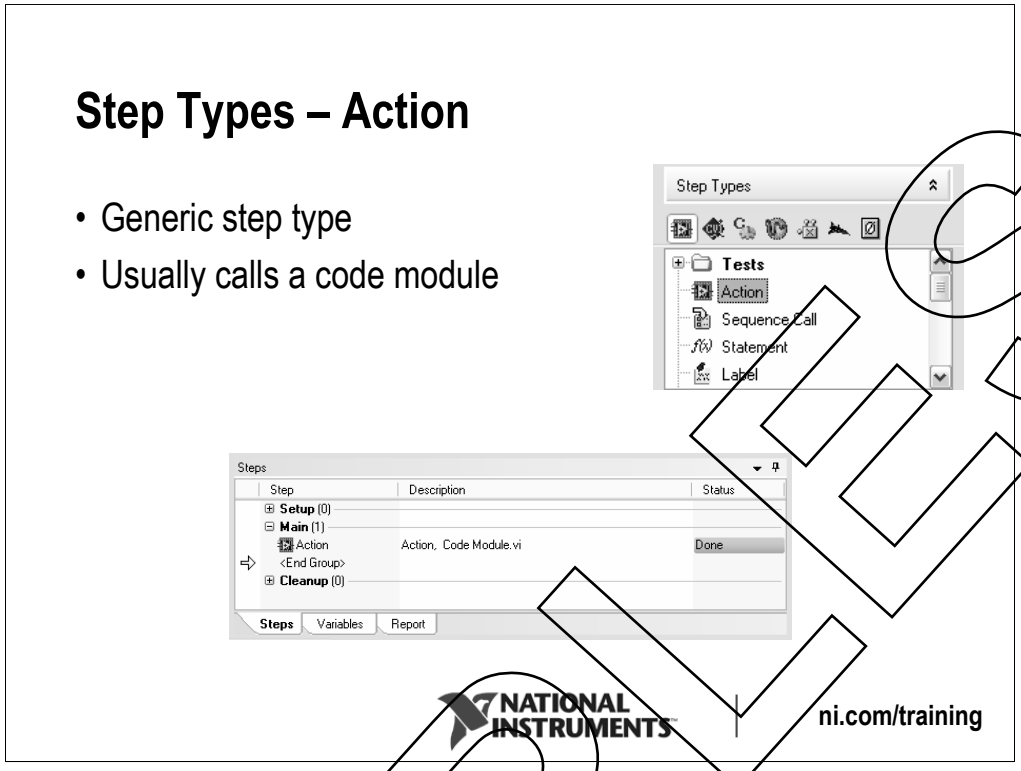
TestStand includes the following built-in step types:

- Tests
- Action
- Sequence Call
- Label
- Message Popup
- Flow Control
- Synchronization
- Other (Statement, Database, Property Loader, Call Executable, FTP Files, LabVIEW Utility, and IVI-C)



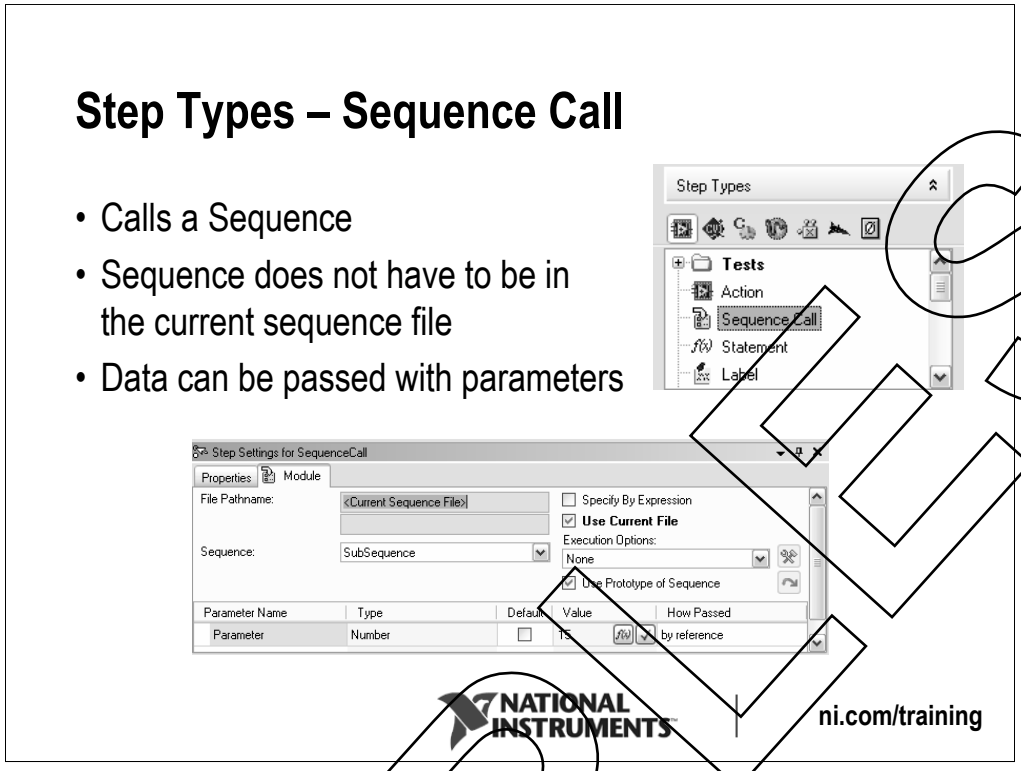
Step Types – Tests

The most essential components of a test sequence are the tests that determine whether a UUT has passed or failed the test sequence. Tests step types represent the tests in your system. Each Tests step typically executes a code module and returns a pass or fail status based on the results of the code. Some tests use the None adapter and execute TestStand statements and expressions to determine the status, rather than calling an external code module. Tests steps can return one of two values for the status step property—Passed or Failed. Refer to the *Step Properties* section for more information on the status step property.



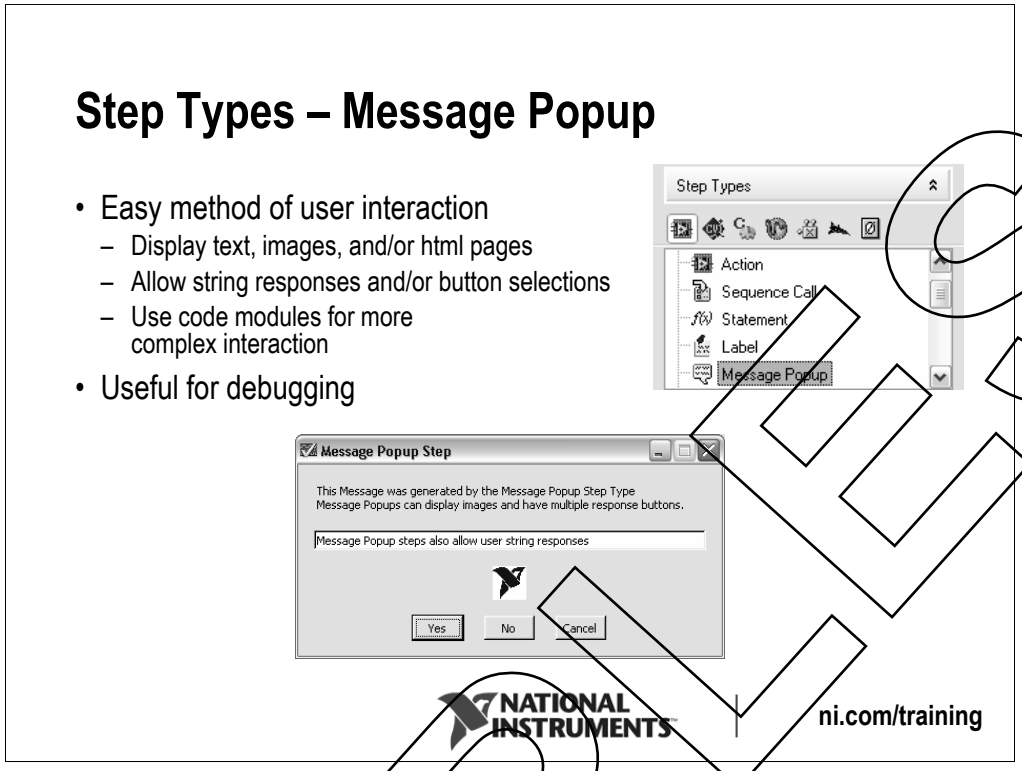
Step Types – Action

Similar to Tests steps, Action steps call a code module. Action steps execute an operation and then complete, rather than return a Passed or Failed status. The Action step is a generic step type you can use in many ways, depending on the code that it calls. Like Tests steps, Action steps can use the None adapter and execute TestStand statements and expressions, rather than call a code module. However, the Statement step type is a more appropriate method of performing actions without code modules.



Step Types – Sequence Call

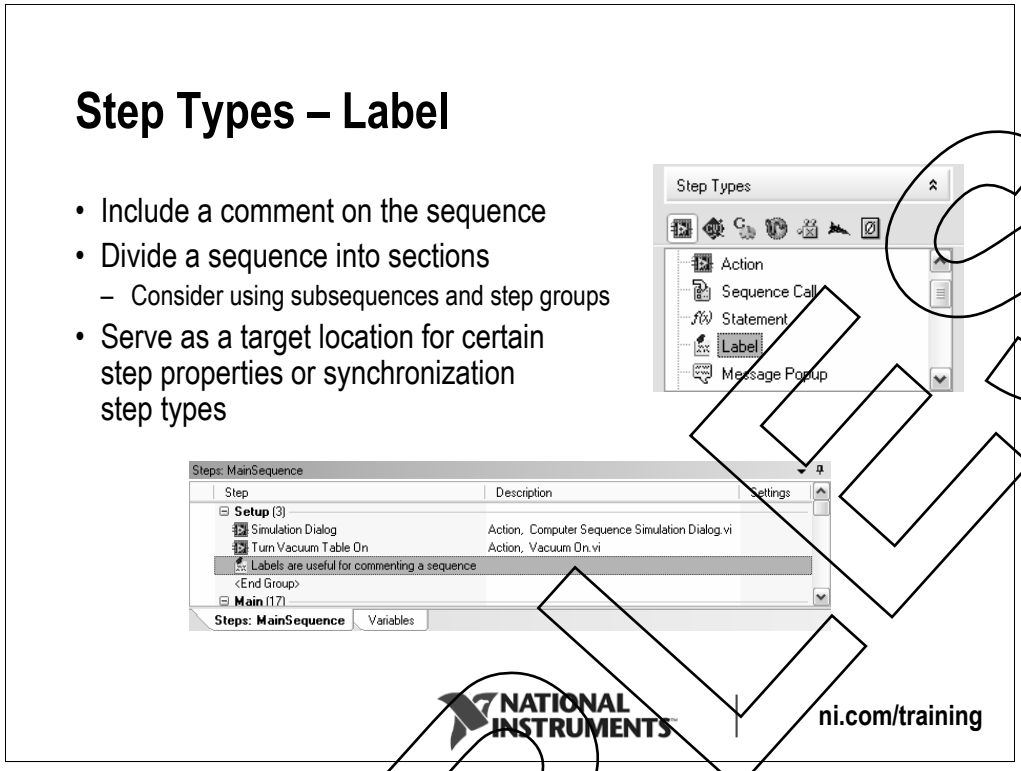
TestStand sequences can call other sequences as subsequences. Use the Sequence Call step type to specify a sequence to execute. You can select the sequence to call from the current sequence file or select a sequence in another sequence file stored on disk. When calling a sequence, you often pass data to the sequence through variables called parameters. Refer to Lesson 5, *Managing Data*, for more information about parameters.



Step Types – Message Popup

Use the Message Popup step type to display a message to the user and, optionally, get simple feedback from the user. The Message Popup step type can display text, an image, and/or a Web page. Use the Message Popup step to get feedback from the user by prompting the user to click one of up to six buttons or enter a string in a response box. For more complex user interaction, use an Action or Tests step type to call a code module written in a language that can create user interfaces.

You can also use the Message Popup step type to help debug a program. Similar to a Watch pane, you can use it to display data or to provide a visual confirmation that a certain section of steps is executing. The built-in debugging tools are more robust, but a message box is simple, and does not require you to halt execution using breakpoints or single stepping.



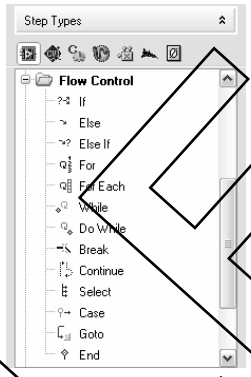
Step Types – Label

Use Label steps types to comment sequences and create reminders or placeholders for other steps. You can also use labels to divide sequences into conceptual chunks. However, keep in mind that subsequences or step groups organize steps more effectively.

Use a Label step as the target for a Goto step. You can use Label steps to rearrange or delete other steps in a sequence without having to change the specification of targets in Goto steps. Label steps do not pass or fail. After a Label step executes, the TestStand Engine sets the step status to Done or Error.

Step Types – Flow Control

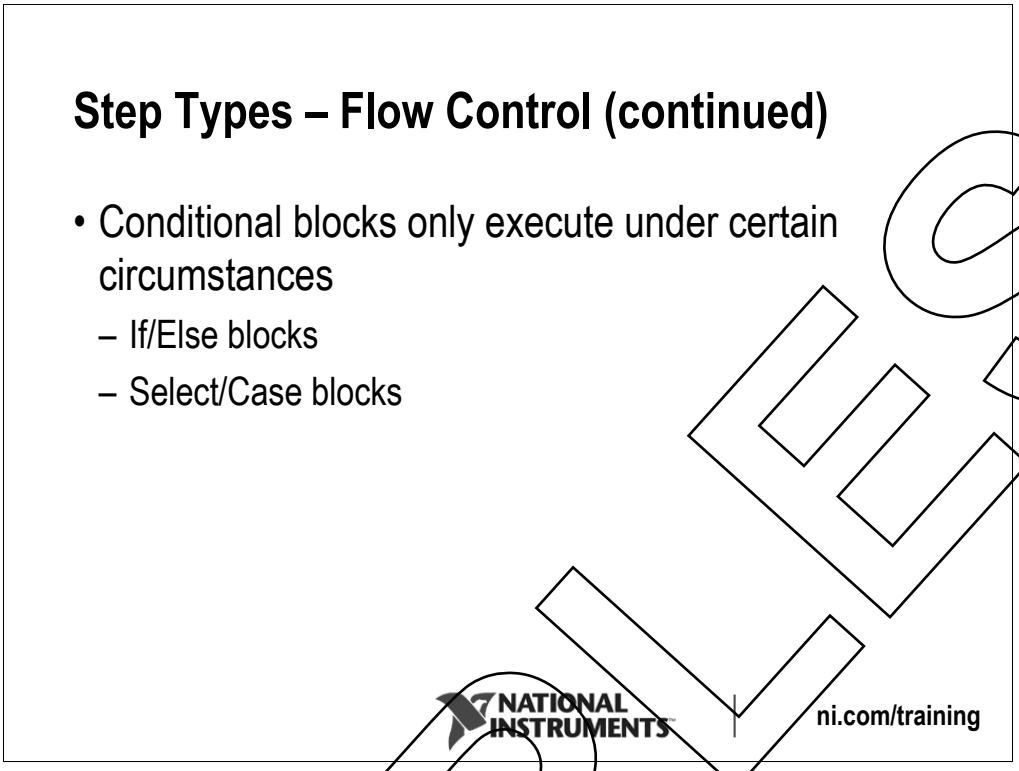
- TestStand normally executes each step in the sequence in order
- Flow Control step types modify this order
- Except for the Goto step, all Flow Control step types define a block of steps that are affected



NATIONAL INSTRUMENTS | ni.com/training

Step Types – Flow Control

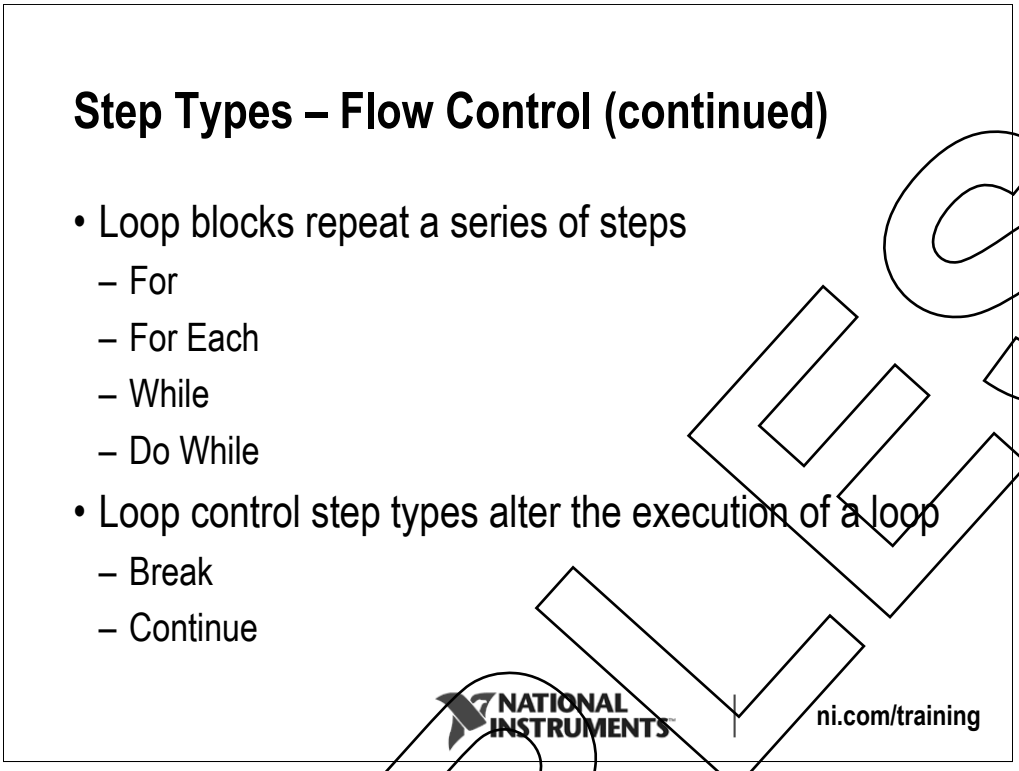
Use Flow Control steps to control execution flow within a sequence. The Steps pane automatically inserts steps that complete the flow control block, such as inserting a Case and End step when you insert a Select step. The Flow Control step defines the beginning of the block. An End step defines the end of the block. The Steps pane also indents flow control blocks and highlights errors in flow control. Refer to the *NI TestStand Help* for more information about the edit tabs for the Flow Control step types.



Step Types – Flow Control (continued)

Conditional Flow Control blocks only execute under certain conditions. Conditional Flow Control step types include the following:

- If
- Else
- Else If
- Select
- Case



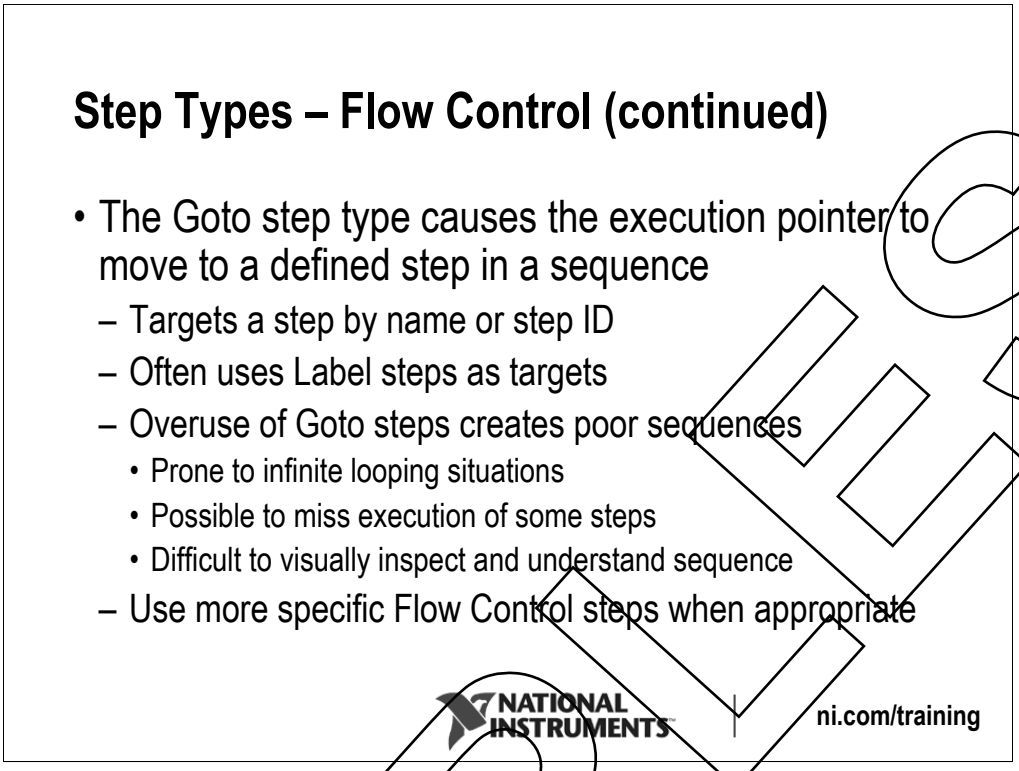
Step Types – Flow Control (continued)

Loop Flow Control blocks repeat a series of steps. Loop Flow Control step types include the following:

- For
- For Each
- While
- Do While

Loop control step types alter the execution of a loop. Loop control step types include the following:

- Break
- Continue

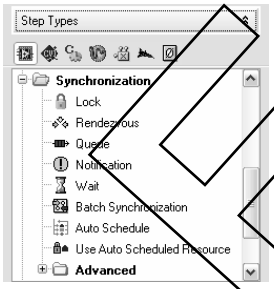


Step Types – Flow Control (continued)

Use Goto steps to set the next step that the TestStand Engine executes. You usually use a Label step as the target of a Goto step, so you can rearrange or delete steps in a sequence without changing the specification of targets in Goto steps.

Step Types – Synchronization

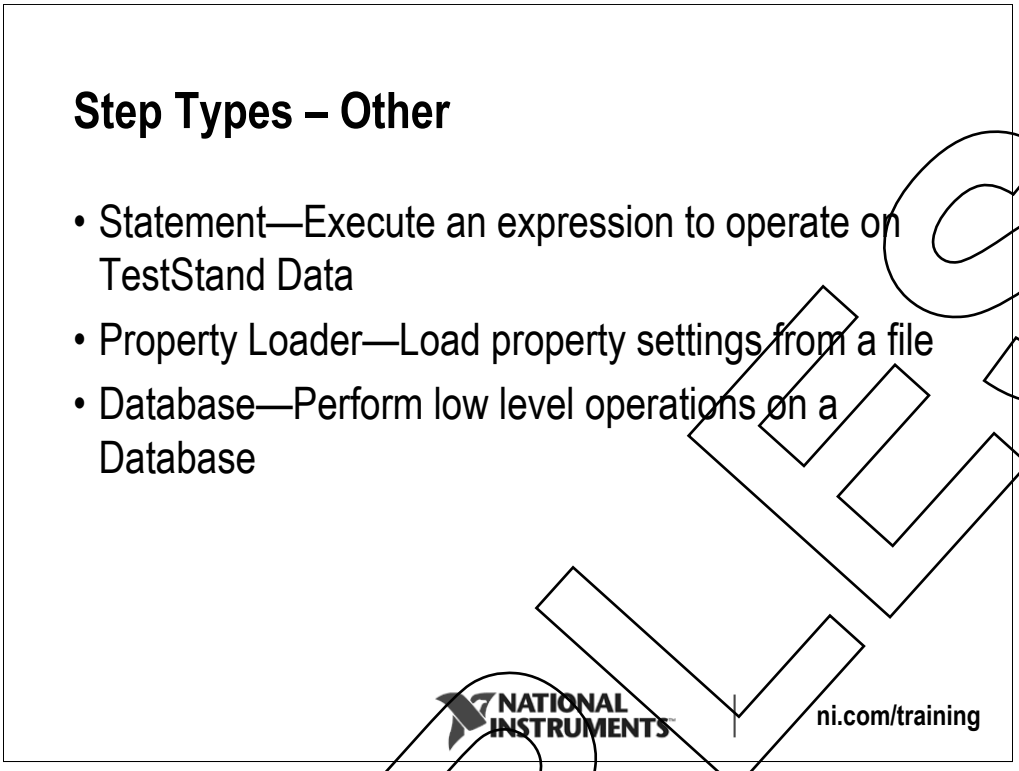
- Wait—pause execution for a specified time
- Other Synchronization steps control and communicate between parallel sequences



NATIONAL INSTRUMENTS | ni.com/training

Step Types – Synchronization

Use Synchronization step types to synchronize, pass data between, and perform other operations in multiple threads of an execution or multiple running executions in the same process. Most Synchronization step types create and control a particular type of Synchronization object. Configure these steps using step type-specific dialog boxes. You do not write code modules for these steps. Refer to Lesson 8, *Executing Tests in Parallel*, for more information about synchronization.



Step Types – Other

The following step types are described in more detail in other locations of the TestStand courses:

- **Statement**—Executes a TestStand expression. Use this step type to alter or set TestStand variable or property values. Refer to Lesson 5, *Managing Data*, for more information on expressions, properties, and variables.
- **Property Loader**—Loads property settings from a file. Refer to Lesson 5, *Managing Data*, for more information about loading property settings.
- **Database**—Directly accesses a database. You do not have to use Database step types to perform database logging. Refer to Lesson 7, *Configuring TestStand*, for more information about database logging. Refer to the *NI TestStand II: Customization* course for more information about the Database step types.

NI TestStand™ I: Introduction Exercises

Course Software Version 4.0
July 2007 Edition
Part Number 324606A-01

Contents

4

Creating Sequences

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 check the volume label of the CD. Before you check the volume label, you should prompt the operator to insert the CD into the drive.

Design

To satisfy the requirements, you must create a new Sequence File and put 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 I\CD Test\Code Modules\ Volume Information.vi`
Comparison Type	CaseSensitive
Expected String Value	`"500815L-01"`
Parameter: Drive Letter	CD Drive letter, such as `"D:"` or `"E:"`
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 I: Introduction Course Manual* for information about installing and using the LabVIEW Runtime Engine.

Implementation

1. Create a new sequence file.
   - ❑ Close any open sequence files in the TestStand sequence editor.
   - ❑ Select **File»New Sequence File**.
   - ❑ Select **File»Save**.
   - ❑ Save the sequence file as `<Exercises>\TestStand I\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.
- ❑ Select the **Text and Buttons** tab in the Step Settings pane.
- ❑ Enter `"Insert CD"` in the Title Expression textbox of the Text and Buttons tab.

Tip 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 of the Text and Buttons tab.
- ❑ The Step Settings pane should be similar to Figure 4-1.

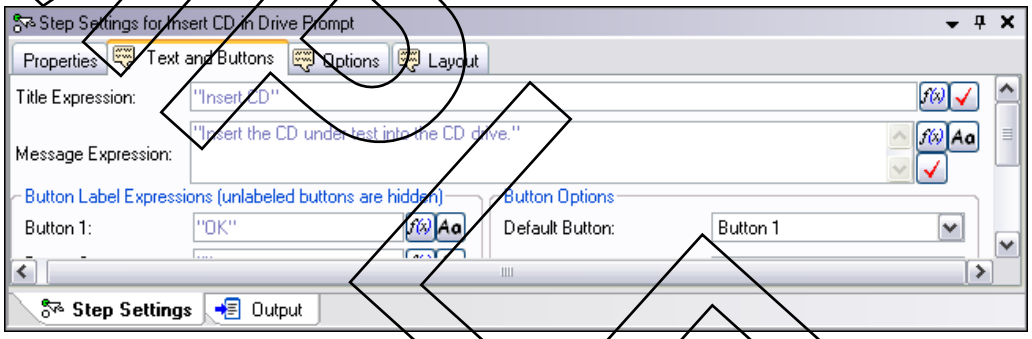


Figure 4-1. Text and Buttons Settings for Insert CD in Drive Prompt Step

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 in 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 in the Step Settings pane.
- ❑ Select **Case Sensitive** from the Comparison Type pull-down menu on the Limits tab.
- ❑ Enter `"500815L-01"` in the Expected String Value textbox on the Limits tab.

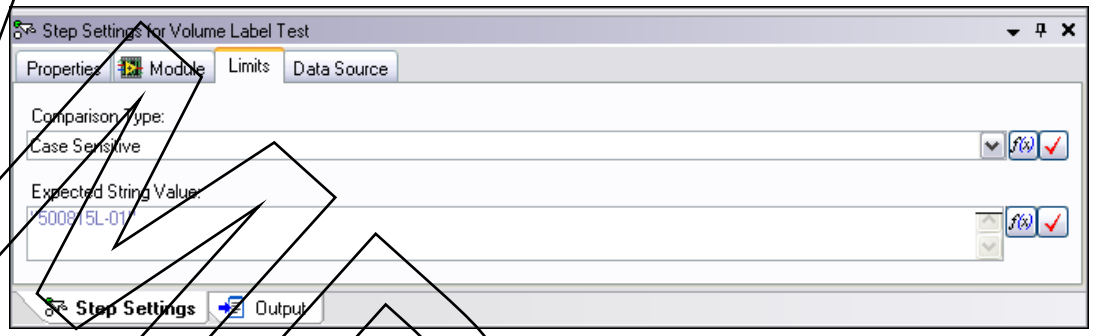


Figure 4-2. Limit Settings for Volume Label Test Step

- ❑ Select the **Module** tab in the Step Settings pane.
- ❑ Click **Browse for VI**.
- ❑ Select `<Exercises>\TestStand I\CD Test\Code Modules\Volume Information.vi`.
- ❑ 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 your files so that you can move the project folder to a new location without breaking the file references.

- ❑ Click **OK**.
- ❑ In the parameters table of the Module tab, disable the **Default** checkbox for the Drive Letter parameter.

- ❑ Enter the letter of your CD drive as the Value for the Drive Letter parameter. For example, enter `"D:"` if your CD drive is the `D:` drive.

Tip You can use Windows Explorer to find your CD drive letter. It is listed in parentheses after the drive name.

- ❑ 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.

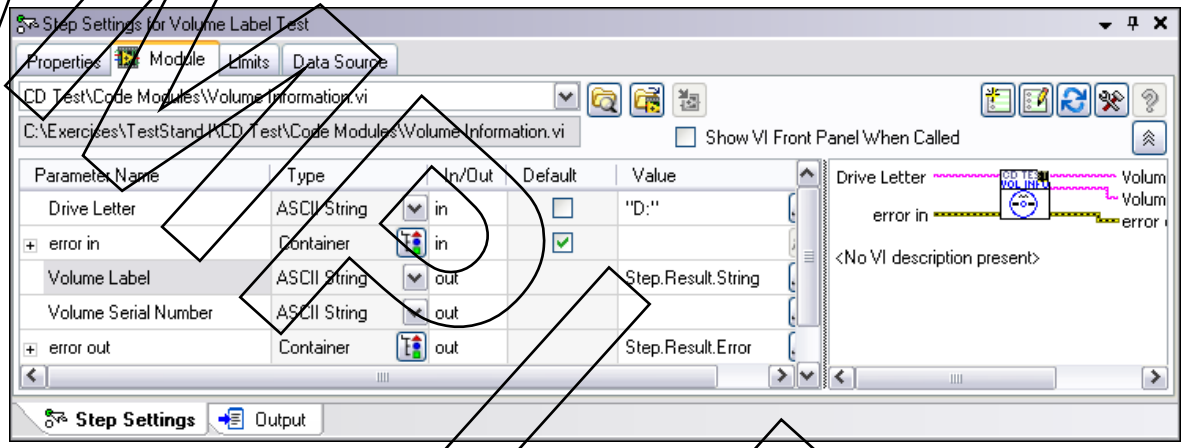


Figure 4-3. Module Settings for Volume Label Test Step

- ❑ Verify that `Step.Result.Error` is the Value for the error out parameter.
- ❑ Verify that your sequence matches Figure 4-4.
- ❑ Select **File»Save**.

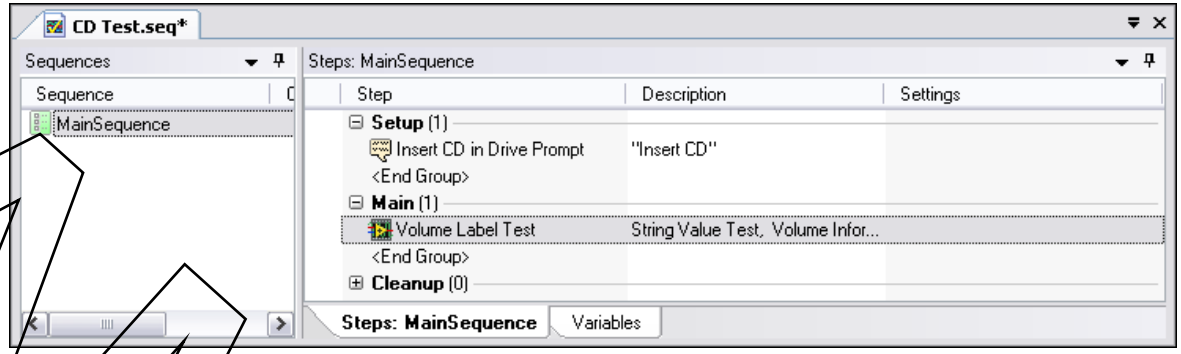


Figure 4-4. CD Test Sequence

Testing

1. Test a passing result.
   - ❑ 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.
2. Test a failing result.
   - ❑ Select **Execute»Single Pass**.
   - ❑ When the Insert CD dialog box appears, place any CD other than the CD from your course manual into the drive.

Tip If you do not have another CD available, you can change the Value of the Drive Letter parameter to the drive letter of your hard drive. This returns the volume label of your hard drive, which creates a failing result. If you use this approach, be sure to set the drive letter back to your CD drive when you finish this step.

- ❑ Click **OK**.
- ❑ When the test report appears, verify that the UUT Result is **`Failed`**.

3. Test with an error result.
   - ❑ Select **Execute»Restart**.
   - ❑ When the Insert CD dialog box appears, remove all disks from the CD drive.
   - ❑ Click **OK**.
   - ❑ When the Run-Time Error dialog box appears, observe the error code that displays. The current error description is not very informative. You will 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 report tab of the Execution pane and select **Close**.

Tip The title of the report tab begins with `Single Pass`.