DOCUMENTATION UPDATES

NATIONAL INSTRUMENTS

This document contains information about the changes and additions to the TestStand documentation.

Changes to the TestStand User Manual

This section contains information about several changes and additions to the *TestStand User Manual*.

Chapter 2—Sequence Editor Options

Make the following modifications to the indicated sections in Chapter 2, *Sequence Editor Options*, in the *TestStand User Manual*.

Workspace Window

The icons in the Workspace window have changed. Replace the section on page 2-9 with the following text:

The following icons appear in the Workspace window.

- Project file.
- Workspace file.
- Sequence file.
- Any file that does not have a file association.
- File is a code module or folder contains code modules.
- File not found on disk.
- File is currently checked into the source code control system.
- File is currently checked out of the source code control system.

Multiline Edit Control in Dialog Boxes

Add the following note to page 2-18, before the *Controlling Sequence Flow* section.

Note In TestStand 2.0.1, press <Enter> in a comment control in any dialog box to close that dialog box. To insert a carriage return in a comment control, press <Ctrl-Enter>.

December 2001

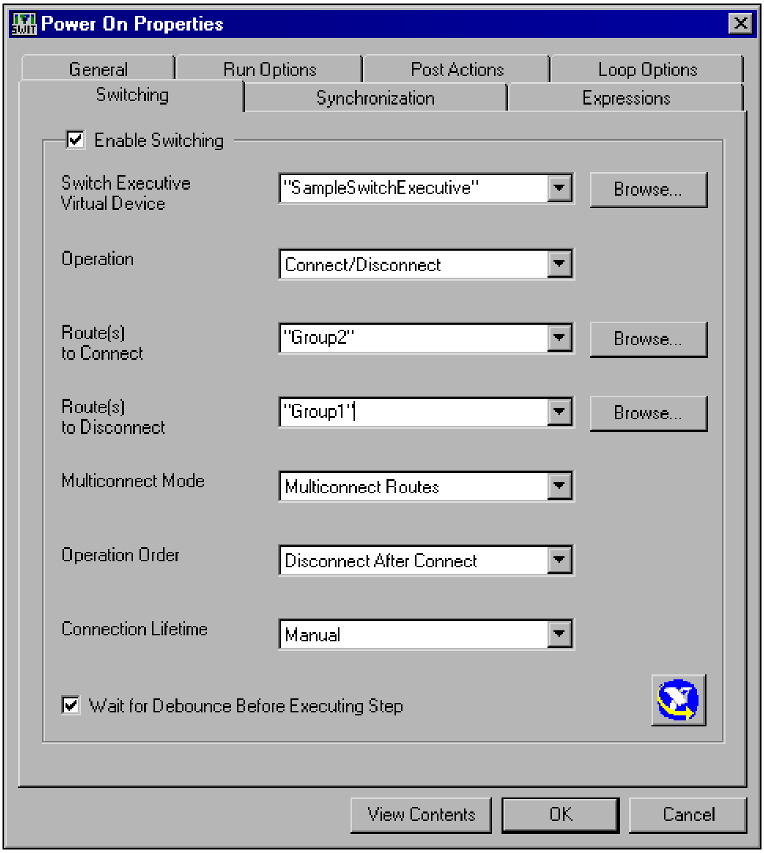
Chapter 5—Step Properties Dialog Box

TestStand 2.0.1 includes a Switching tab on the Step Properties dialog box. Use this feature with National Instruments Switch Executive. Add the following text to page 5-31, after the *Expressions Tab* section.

Switching Tab

Use the Switching tab to specify the switching action you want TestStand to perform for a step.

Figure 1 shows the Switching tab on the Step Properties dialog box.



☑ Enable Switching

Switch Executive Virtual Device: "SampleSwitchExecutive"

Operation: Connect/Disconnect

Route(s) to Connect: "Group2"

Route(s) to Disconnect: "Group1"

Multiconnect Mode: Multiconnect Routes

Operation Order: Disconnect After Connect

Connection Lifetime: Manual

☑ Wait for Debounce Before Executing Step

Figure 1. Switching Tab on the Step Properties Dialog Box

The Switching tab contains the following controls:

- **Enable Switching**—Specifies whether to perform the specified switching action for the step.
- **Switch Executive Virtual Device**—Specifies an expression that TestStand evaluates at run time to determine the virtual device on which TestStand performs the switching action.
- **Operation**—Specifies whether to connect or disconnect the specified routes or to disconnect all previously connected routes. This operation returns as soon as the instrument is ready for another operation, which may occur before or after the switches involved settle. Use the **Wait For Debounce** option in the **Operation Order** ring control if you want to wait until all switches have debounced. In the **Operation** control, you can choose from the following options:
  - **Connect**—Connects the paths for the specified routes in the **Route(s) to Connect** control.
  - **Disconnect**—Disconnects the paths defined by the specified routes in the **Route(s) to Disconnect** control.
  - **Disconnect All**—Disconnects all paths previously created.
  - **Connect/Disconnect**—Connects the paths for the routes specified in the **Route(s) to Connect** control and disconnects the paths for the routes specified in the **Route(s) to Disconnect** control. Use the **Operation Order** control to specify whether the disconnect operation occurs before or after the connect operation.
- **Route(s) to Connect**—Specifies the routes you are connecting. The expression must be a valid route specification string, as defined by the National Instruments Switch Executive configuration for the virtual device name you are using. The string can be a combination of route group alias names, route names, and physical route paths.
- **Route(s) to Disconnect**—Specifies the routes you are disconnecting. The expression must be a valid route specification string, as defined by the National Instruments Switch Executive configuration for the virtual device name you are using. The string can be a combination of route group alias names, route names, and physical route paths.

Note For more information on the route specification string, refer to `<TestStand>\Doc\IVIStepTypes.pdf`.

- **Multiconnect Mode**—Specifies the behavior when more than one connection operation occurs on a specific route. You can choose from the following options:
  - **Multiconnect Routes**—A route can be connected multiple times. The route must contain the same endpoints and path. National Instruments Switch Executive automatically reference counts the

routes. If you issue multiple connect operations for a specific route, the route is not physically disconnected until an equal number of disconnect operations occur. You can either issue the disconnect operation manually or use a lifetime setting for the route. The **Disconnect All** operation disconnects a route even if the route reference count is greater than 1.

  - **No Multiconnect**—A route can only be connected once. Any attempt to reconnect a route that is already connected results in an error.

- **Operation Order**—Specifies whether the disconnect operation occurs before or after the connect operation. You can choose from the following options:
  - **Disconnect Before Connect**—Disconnects the specified routes before connecting any routes. This is the typical mode of operation.
  - **Disconnect After Connect**—Connects the specified routes before disconnecting any routes. Use this mode of operation when you are switching electric current and want to ensure that a load is always connected to your source.
- **Connection Lifetime**—Specifies the lifetime that TestStand applies to the routes specified for the connect or connect/disconnect actions. You can specify whether you want the route to exist until manually disconnected later, or until the step, sequence, thread or execution completes. If you use the multiconnect route mode, a route can exist longer if another step specifies its own lifetime for the same route. Selecting a lifetime other than manual guarantees that the route stays connected as long as the step, sequence, thread, or execution in which you connect is still executing. If you manually disconnect a route that was previously connected using a non-manual lifetime setting, TestStand releases the reference to the route for the last step that performed a connect action for that route.
- **Wait for Debounce**—Specifies whether the operation waits for all switches to debounce before returning to TestStand. The wait for debounce occurs at different times depending on what operation you specify. For the Connect, Disconnect, and Disconnect All operations, wait for debounce occurs after the operation. For the Connect/Disconnect operation, wait for debounce occurs after the connect action.

Chapter 6—Sequence Execution

Make the following modifications to the indicated sections in Chapter 6, *Sequence Execution*, in the *TestStand User Manual*.

Step Execution

Table 6-5 on page 6-25 has changed. Replace the table and text with the following:

Table 6-5. Order of Actions That a Step Performs

Action Number	Description	Remarks
1	Allocate step result in Locals.ResultList	—
2	Enter batch synchronization section	If option is set
3	Check run mode for `Skipped`	Skip to action 31 if `Skipped`
4	Evaluate precondition	Skip to action 31 if `False`
5	Acquire step lock	If option is set
6	Check run mode for `Force Pass` and `Force Fail`	Skip to action 30 if `Force Pass` or `Force Fail`
7	Load module if not already loaded	—
8	Perform switching operation	If option is set
9	Evaluate Loop Initialization expression	Only if looping
10	Evaluate Loop While expression	Only if looping. Skip to action 24 if `False`
11	Allocate loop iteration result	Only if looping
12	Call Pre Step engine callbacks	—
13	Evaluate Pre expression	—
14	Call Pre Step substeps for step type	—
15	Call module	—

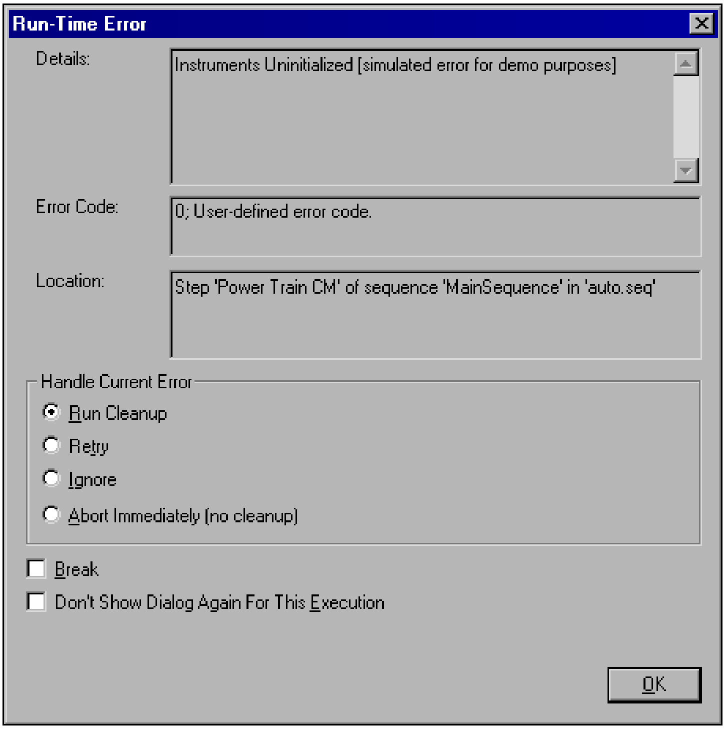
Table 6-5. Order of Actions That a Step Performs (Continued)

Action Number	Description	Remarks
16	Call Post Step substeps for step type	TestStand calls Post Step substeps even if the user code module generates a run-time error. This enables Post Step substeps to perform error handling, if appropriate.
17	Evaluate Post expression	—
18	Evaluate Status expression	—
19	Call Post Step engine callbacks	—
20	Call Post Step failure engine callbacks	Only if loop iteration fails
21	Fill out loop iteration result	Only if looping
22	Call PostResultListEntry engine callbacks	Only if looping
23	Evaluate Loop Increment expression, return to action 10	Only if looping
24	Evaluate Loop Status expression	Only if looping
25	Perform Disconnect switching operation for routes using step lifetime	If option is set
26	Unload module if required	—
27	Update sequence failed state	—
28	Call Post Step failure engine callbacks	Only if step fails
29	Execute post action	—
30	Release step lock	If option is set
31	Exit batch synchronization section	If option is set
32	Fill out step result	—
33	Call PostResultListEntry engine callbacks	—

Usually, a step performs only a subset of these actions, depending on the configuration of the step and the test station. When TestStand detects a run-time error, it usually proceeds to action 30. If a run-time error occurs in a loop iteration, TestStand performs action 20 before it performs action 30.

Run-Time Errors

Figure 6-9 and the text on page 6-30 has changed. Make the following modifications to the Run-Time Error dialog box section.



Run-Time Error

Details: Instruments Uninitialized [simulated error for demo purposes]

Error Code: 0; User-defined error code.

Location: Step 'Power Train CM' of sequence 'MainSequence' in 'auto.seq'

Handle Current Error

● Run Cleanup

○ Retry

○ Ignore

○ Abort Immediately (no cleanup)

□ Break

□ Don't Show Dialog Again For This Execution

Figure 6-9. Run-Time Error Dialog Box

The Run-Time Error dialog box gives you four possible ways to handle the run-time error. Any option you select under the **Handle Current Error** section applies to all future run-time errors for the current run of the executions.

- **Run Cleanup**—Execution proceeds to the Cleanup step group for the sequence. Run Cleanup is the default action when you disable the **Show Dialog On Run-Time Error** option.
- **Retry**—TestStand runs the step again.
- **Ignore**—TestStand does not set the internal status of the sequence to `Error`. Instead, TestStand resets the `Error.Occurred` property of

the step to `False` and execution continues normally with the next step in the sequence. The `Result.Status` property of the step remains set to `Error`.

- **Abort Immediately**—TestStand stops execution immediately, without running any cleanup steps.
- **Break**—When you choose the **Run Cleanup** or **Ignore** actions and enable the **Break** option, TestStand suspends execution at the step that caused the run-time error. The **Break** option dims if you choose **Abort Immediately**.
- **Don't Show Dialog Again For This Execution**—Prevents the Run-Time Error dialog box from appearing for any run-time errors that occur later in the execution.
- **Don't Show Again For This Batch**—When you run an execution using the Batch process model, TestStand enables this checkbox. This option prevents the Run-Time Error dialog box from appearing for any errors that occur in later executions that are part of the same batch as the current execution.

Chapter 8—Sequence Context and Expressions

Add the following text to the *Sequence Context Subproperties* section on page 8-3.

When an execution completes or a thread that you launch from a Sequence Call step finishes, TestStand updates the following subproperties of the Execution or Thread object from the top level sequence that the execution or thread runs.

- `Result.Error.Code`, `Result.Error.Msg`, `Result.Error.Occurred`—Contains any error information from the sequence.
- `Result.SequenceFile`—The absolute path of the sequence file that contains the sequence.
- `Result.Sequence`—The name of the top level sequence that the execution or thread runs.
- `Result.Status`—The status of the sequence.
- `Result.ResultList`—The value of `Locals.ResultList` for the sequence. This property contains the results for the steps that the sequence ran.

Chapter 11—Synchronization Step Types

Make the following modifications to the indicated sections in Chapter 11, *Synchronization Step Types*, in the *TestStand User Manual*.

Retrieving the Results from Executions and Threads

Replace the second paragraph in the *Retrieving the Results from Executions and Threads* section on page 11-52 with the following paragraph:

In addition to updating the Status and Error properties of the Wait step, TestStand also records the results of the execution or thread in the result list entry for the Wait step in the same way that it records the results of a synchronous sequence call in the result list entry for a Sequence Call step. The default report generator and database logger process a Wait step result as if the Wait step called the asynchronous sequence. To record the results of the asynchronous sequence in the result list entry for the Wait step, TestStand creates a `TS.SequenceCall` property. Refer to the *Subsequence Results* section of Chapter 6, *Sequence Execution*, for a description of the subproperties that TestStand creates with the `TS.SequenceCall` property.

One-Thread-Only Sections

Add the following sentence to the *One-Thread-Only Sections* section on page 11-57.

If a thread generates a run-time error in a one-thread-only section, TestStand generates a run-time error in each remaining thread in the batch. In this way, a run-time error in an operation that applies to the entire batch results in a run-time error in each thread in the batch.

Mismatched Sections

Add the following note to the *Mismatched Sections* section on page 11-58.

Note Typically, you set the Batch Synchronization ring control on the Synchronization tab of the Step Properties dialog box to Parallel for any step that can cause a branch to a previous step, for example, a Goto step or a step with a Goto post action. This ensures that the branching step executes concurrently for all sequence instances, even when the step is nested within another synchronized section. Executing the backward branch concurrently prevents unintended, mismatched sections that can result from one thread reaching a previous section while another thread that has yet to branch back waits at a different section before the branch.

Chapter 13—Module Adapters

Make the following modifications to the indicated sections in Chapter 13, *Module Adapters*, in the *TestStand User Manual*.

Sequence Adapter

Replace the third paragraph in the *Sequence Adapter* section on page 13-40 with the following paragraph:

After the Sequence Call step executes, the sequence adapter may set the step status. If the sequence that the step calls fails, the sequence adapter sets the step status to `Failed`. If the sequence does not fail and no run-time error occurs, the sequence adapter does not set the step status. Depending on the type of step, the resulting status is `Done` or `Passed`. If a run-time error occurs in the sequence, the sequence adapter sets the step status to `Error` and sets the `Result.Error.Occurred` property to `True`. The sequence adapter also sets the `Result.Error.Code` and `Result.Error.Msg` properties to the values of the same properties in the subsequence step that generated the run-time error.

Specifying a Sequence Adapter Module

Replace the note in the *Specifying a Sequence Adapter Module* section on page 13-42 with the following note:

Note When you specify that a sequence runs in a new thread or a new execution, the status of the Sequence Call step is `Done` or `Error` and does not depend on the status of the sequence you call. In addition, the Sequence Call step does not record the results of the steps in the asynchronous sequence. You can use a Wait step to obtain the status, error information, and results of an asynchronous sequence call. If you do not use a Wait step and you specify in the sequence call that TestStand waits for the asynchronous call to complete at the end of the current sequence, TestStand applies a `Failed` status or run-time error in the asynchronous sequence to the calling sequence. In this case, TestStand discards the step results from the asynchronous sequence and the report displays the failure or error in the result for the step that calls the calling sequence.

Execution Settings

Under the **Wait for Execution to Complete** bullet on page 13-46, replace the option **At end of current sequence** with the following:

- **At end of current sequence**—The calling sequence waits for the execution to complete before the calling sequence returns. When you specify this option, the sequence adapter attaches the result list for the initial sequence in the execution to the result for the

Sequence Call step. The sequence adapter also copies the execution Status and Error properties to the sequence call Status and Error properties.

Chapter 15—Managing Reports

Make the following modifications to the indicated sections in Chapter 15, *Managing Reports*, in the *TestStand User Manual*.

Report File Pathname Tab

Some of the options on the Report File Pathname tab have changed. Make the following modifications to the indicated sections.

On page 15-12, replace the **Append UUT Serial Number to Report File Name** option with the following description.

- **New UUT Report for Each UUT**—Appends the UUT serial number to the report file name. For example, `auto_report.html` might become `auto_report[ABC12345].html`. This option causes the report generator to create a separate file for each UUT.

On page 15-12, replace the **Append Batch Serial Number to UUT Report File Name** option with the following description.

- **New UUT Report File for Each Batch**—Appends the batch serial number to the UUT report file name. This option prevents the report generator from storing in the same file UUT reports from different batches. This control is available only when you use the Batch process model.

On page 15-12, replace the **Append Test Socket Index to UUT Report File Name** option with the following description.

- **New UUT Report File for Each Test Socket**—Appends the test socket index number to the UUT report file name. This option prevents the report generator from storing in the same file UUT reports from different test sockets. This control is available only when you use the Batch or Parallel process model.

On page 15-13, replace the **Append Batch Serial Number to Batch Report File Name** option with the following description.

- **New Batch Report File for Each Batch**—Appends the batch serial number to the batch report file name. This option prevents the report generator from storing in the same file batch reports from different batches. This control is available only when you use the Batch process model.

Chapter 18—Databases

Make the following modifications to the indicated sections in Chapter 18, *Databases*, in the *TestStand User Manual*.

Columns/Parameters Tab

On page 18-19, replace the description for the **Direction** ring control with the following description:

- **Direction**—Specifies whether the column or parameter is an input or output value. You can select one of the following options:
  - **Input**—Indicates that TestStand writes the column value or parameter to the database.
  - **Output**—Indicates that TestStand retrieves the column value or parameter from the database. This option is only valid for statements of type Stored Procedure.
  - **Input/Output**—Indicates that TestStand writes and retrieves the column value or parameter. This option is only valid for statements of type Stored Procedure.
  - **Return Value**—Indicates that TestStand retrieves the parameter value as a return value from the database. This option is only valid for statements of type Stored Procedure.

File Menu

Add the following bullet below and at the same level as New Data Link in the *File Menu* section on page 18-33.

- **Save Data Link**—Prompts the user for a filename and saves the data link of the active window to a `.udl` file. This option is only visible when a TableView or Execute SQL window is active.

SQL Statement Tab

On page 18-53, replace the description for the **Number of Records Selected** control with the following description.

- **Number of Records Selected**—Specifies a variable or property to which the step assigns the number of records that the SQL statement returns. The variable or property is of the Number data type. The step returns a value of `-1` when the step is unable to determine the number of records returned. This typically occurs when the Advanced tab specifies the cursor location as server or when the step specifies to use the default value and the provider defaults to a server cursor.

Changes to the TestStand Programmer Help

Make changes to the following topics in the *TestStand Programmer Help*.

StepTypes Constants

The StepTypes constant StepType_LimitLoader with a value of “LimitLoaderStep” is obsolete. Use the new constant StepType_PropertyLoader which has a value of “NI_VariableAndPropertyLoader”.

Expressions

The example sequence file in this section uses the Property Loader step rather than the Limit Loader step. The Limit Loader step is obsolete.

Loading Limits

Each example sequence file in this section uses the Property Loader step rather than the Limit Loader step. The Limit Loader step is obsolete.