SystemTest™ Release Notes

MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

SystemTest™ Release Notes

Trademarks

Patents

Contents

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2009a

R2008b

R2008a

R2007b

R2007a

R2006b

R2006a+

R2014b

Version: 2.6.8

No New Features or Changes

R2014a

Version: 2.6.7

Bug Fixes

R2013b

Version: 2.6.6

Bug Fixes

R2013a

Version: 2.6.5

No New Features or Changes

R2012b

Version: 2.6.4

No New Features or Changes

R2012a

Version: 2.6.3

No New Features or Changes

R2011b

Version: 2.6.2

Bug Fixes

R2011a

Version: 2.6.1

New Features

Bug Fixes

Compatibility Considerations

Deprecation of Test Results Viewer

In R2010b, the Test Results Viewer was no longer available from the SystemTest™ desktop. It could still be opened via the `stviewer` function.

The following were removed from the desktop user interface in R2010b:

- The **Visualize and plot saved results by launching the Test Results Viewer** option on the **General** tab of the **Test Properties** pane.
- The **Tools > Test Results Viewer** menu command.
- In the Preferences dialog box, the confirmation warning “Warn about launching the Test Results Viewer without any saved results.”

In R2011a, the Test Results Viewer is no longer available, even via the `stviewer` function. We recommend using `stresults` to view saved results instead. You can view the results you have chosen to save for your test using a workspace variable called `stresults`. It provides access to the test results object, which is useful for comparing the results of separate test runs and for postprocessing test results. For more information on viewing saved results see Accessing Test Results from the MATLAB Command Line.

For visualization of data within the SystemTest software, you can use the General Plot element and view the generated plots after a test run. For information on using the General Plot element, see General Plot Element.

For visualizing test cases and signals, you can use the Test Case Editor. For more information, see Authoring Signals in the Test Case Editor.

Compatibility Considerations

When you open a test that was previously configured to use the Test Results Viewer, you will get a warning dialog saying “This test was configured to open the Test Results Viewer when the test finishes running. This option is no longer available.”

The following outline describes the timing of the deprecation process:

- **R2010b** — The Test Results Viewer was no longer available from the SystemTest desktop. It could still be opened via the `stviewer` function.
- **R2011a (current release)** — The Test Results Viewer has been removed. The `stviewer` function produces an error stating that the Viewer is no longer available.
- **R2011b (or later)** — The `stviewer` function will be removed.

R2010b

Version: 2.6

New Features

Bug Fixes

Compatibility Considerations

Automatically Generate SystemTest Test from Simulink

You can now create a SystemTest test from Simulink®. It will automatically generate relevant parts of the test based on your model. The feature automatically generates the following in your test:

- Simulink element, configured properly
- Test Case Data test vector, configured properly
- Signals in the Test Case Editor, configured properly
- Mappings of the Inports in your model

You can also automatically generate the test harness via the MATLAB® command line, using new functions for this feature.

For more information, see Generating a SystemTest Test Harness from a Simulink Model.

Simulink Bus Support in Test Case Editor

The Test Case Editor now supports the use of buses in your model and can be used to create buses and signals within buses. The full bus hierarchy is now reflected in the Test Case Editor and you can work with the signals inside buses.

For more information, see Buses in the Test Case Editor

Additional Model Coverage Support

Two metrics have been added to the Model Coverage support in the Simulink element – **Signal Size Coverage** and **Simulink Design Verifier**. **Signal Size Coverage** records the minimum, maximum, and allocated size for all variable-size signals in a model. The **Simulink Design Verifier** coverage option collects model coverage data for some Simulink Design Verifier™ blocks.

Support for External Embedded MATLAB® files was also added. It enables coverage for external functions that Embedded MATLAB functions call in your model. The Embedded MATLAB functions may be defined in an Embedded MATLAB block.

For more information, see Simulink Model Coverage.

Programmatic Interface for Test Case Editor

The functionality of the Test Case Editor is now available via the MATLAB command line. The new functions will allow you to do the following tasks programmatically:

- create test cases, signals, and segments
- add test cases, signals, and segments to a SystemTest test
- configure and edit test cases, signals, and segments in a test
- extract test cases, signals, and segments from a test

For more information, see Test Case and Signal Authoring Programmatically

Programmatic Interface for Requirements Linking in Test Case Editor

The functionality of requirements linking using Telelogic® DOORS® and the Test Case Editor is now available via the MATLAB command line. The new functions will allow you to do the following tasks programmatically:

- create requirements links to Telelogic DOORS objects
- add requirements to test cases in the Test Case Editor

For more information, see Create Requirements Programmatically

Deprecation of Test Results Viewer

In R2010b, the Test Results Viewer is no longer available from the SystemTest desktop. It can still be opened via the `stviewer` function.

The following have been removed from the desktop user interface:

- The **Visualize and plot saved results by launching the Test Results Viewer** option on the **General** tab of the **Test Properties** pane.
- The **Tools > Test Results Viewer** menu command.
- In the Preferences dialog box, the confirmation warning “Warn about launching the Test Results Viewer without any saved results.”

To open the Test Results Viewer, use the `stviewer` function in MATLAB after a test run.

The Test Results Viewer will no longer be available in a future release, even via the `stviewer` function. We recommend using `stresults` to view saved results instead. You can view the results you have chosen to save for your test using a workspace variable called `stresults`. It provides access to the test results object, which is useful for comparing the results of separate test runs and for postprocessing test results. For more information on viewing saved results see Accessing Test Results from the MATLAB Command Line.

For visualization of data within the SystemTest software, you can use the General Plot element and view the generated plots after a test run. For information on using the General Plot element, see General Plot Element.

For visualizing test cases and signals, you can use the Test Case Editor. For more information, see Authoring Signals in the Test Case Editor.

Compatibility Considerations

When you open a test that was previously configured to use the Test Results Viewer, you will get a warning dialog saying “This test was configured to open the Test Results Viewer when the test finishes running. This option is no longer available.”

If you do open the Viewer using the `stviewer` function, you will get a warning on the command line saying “The viewer will be removed in a future release.”

The following outline describes the timing of the deprecation process:

- **R2010b** — The Test Results Viewer is no longer available from the SystemTest desktop. It can still be opened via the `stviewer` function.

  You will receive the warnings described above.
- **R2011a** — The Test Results Viewer will be removed. The `stviewer` function will produce an error stating that the Viewer is no longer available.
- **R2011b (or later)** — The `stviewer` function will be removed.

Plot Element Transition, Conversion Utility

The Vector Plot and Scalar Plot elements have been replaced by the General Plot element. The General Plot element supports all MATLAB plot types. You can plot any type of data, using any MATLAB plot types or Simulink data.

The General Plot element has special support for the many time-based data formats produced from a Simulink model. The supported data types include an array, a structure, a structure with time, or a time series. You can also plot Simulink data provided by test vectors, such as the Signal Builder Block test vector, the Simulink Design Verifier Data File test vector, or the Spreadsheet Data test vector.

Note: As of R2010a, you could no longer create new Scalar Plot or Vector Plot elements. They no longer appear in the **Insert** menu or the **New Element** button.

Conversion Utility

Tests containing Scalar Plot or Vector Plot elements will not automatically load with those elements. You will be prompted to convert them to General Plot elements.

For information on performing the conversion, and conversion details about the Scalar Plot and Vector Plot elements, see Deprecated Elements.

Compatibility Considerations

For an outline of the timing for this deprecation, see Compatibility Considerations in the R2010a section of the release notes.

Future Deprecation of Hardware Elements

In a future release, possibly R2011a, the following elements will be removed from SystemTest software:

- **Data Acquisition > Analog Input**
- **Data Acquisition > Analog Output**
- **Data Acquisition > Digital Input**
- **Data Acquisition > Digital Output**
- **Image Acquisition > Video Input**
- **Instrument Control > From Instrument**
- **Instrument Control > Query Instrument**
- **Instrument Control > To Instrument**

These elements will be removed in a future release (R2011a or later). We therefore recommend not creating new tests using these elements.

Future Deprecation of Macintosh Support

In a future release, (R2011a or later), the SystemTest software will no longer be supported on any Mac platform, including maci64.

R2010a

Version: 2.5

New Features

Bug Fixes

Compatibility Considerations

Test Case Editor and Signal Authoring

You can create signals in the SystemTest software and use them to test a Simulink model. The Test Case Editor provides a graphical way of creating, editing, and visualizing signal data in SystemTest.

The tool provides the ability to create signals based on commonly used patterns and to specify values for attributes of those signal segments. You can easily create the following types of signals: constant, step, ramp, pulse, square, sine, and custom.

The tool also allows you to view and manage signals, and to organize them into test cases. You can manage a large number of test cases and signals. The Test Case Editor is accessed through the Test Case Data test vector in the SystemTest software.

For information on using the Test Case Editor, see Authoring Signals in the Test Case Editor.

Support for Requirements Linking

You can link test cases created in the new Test Case Editor to requirements that are in Telelogic DOORS. This is done through the **Requirements** tab in the Test Case Editor. The integration allows you to easily link any DOORS requirements to any test cases by selecting them by their object headings.

For more information on linking requirements to test cases, see Link to Requirements in Telelogic DOORS

Test Property Enhancements

Further enhancements were added to saving of results files in the **Test Properties** pane. New output folder options for the test results and the Test Report allow you to select between several relative locations, which makes moving or sharing files easier, or to assign an absolute location.

The options for results files are now on a new tab, **Output Files**, on the **Test Properties** pane.

For more information see Saving Test Results and Generate a Test Report, or the Desktop Help for the **Output Files** tab.

Plot Element Transition, Conversion Utility

The Vector Plot and Scalar Plot elements have been replaced by the General Plot element that was introduced in R2008b. The General Plot element supports all MATLAB plot types. You can plot any type of data, using any MATLAB plot types or Simulink data.

The General Plot element has special support for the many time-based data formats produced from a Simulink model. The supported data types include an array, a structure, a structure with time, or a time series. You can also plot Simulink data provided by test vectors, such as the Signal Builder Block test vector, the Simulink Design Verifier Data File test vector, or the Spreadsheet Data test vector.

Note: You can no longer create new Scalar Plot or Vector Plot elements. They no longer appear in the **Insert** menu or the **New Element** button.

Conversion Utility

Tests containing Scalar Plot or Vector Plot elements will not automatically load with those elements. You will be prompted to convert them to General Plot elements.

For information on performing the conversion, and conversion details about the Scalar Plot and Vector Plot elements, see Deprecated Elements.

Compatibility Considerations

The following outline describes the timing of the transition process:

- **R2009a and R2009b** — You should no longer create new Vector Plot or Scalar Plot elements. Tests containing the old elements continued to work, but we recommended that you do not create any new ones, and instead use the General Plot element.
- **R2010a (current version)** — You can no longer create new Vector Plot or Scalar Plot elements. They no longer appear in the **Insert** menu or the **New Element** button. A warning will appear if you load a test containing the old elements. Tests containing the old elements will continue to work, but you cannot create any new ones.

  You will not lose the use of the old elements in existing tests. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element. See the previous section.

- **A future release** — You can no longer load tests containing Vector Plot or Scalar Plot elements. If you try to load one, you will get an error, and will have to convert the element to the General Plot element. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element.

R2009b

Version: 2.4

New Features

Bug Fixes

Compatibility Considerations

Test Vector Data Viewer

You can now view the test vector data while configuring a Probability Distribution test vector. The viewer will display non-time-based data for this type of test vector, allowing you to view your data before running a test.

Probability Distribution test vector data is displayed in a histogram. To access the data viewer, click the **View Data** button in the Insert Test Vector dialog box.

For more information see Viewing Data While Configuring the Test Vector.

Test Property Enhancements

Enhancements were added to saving of results files in the **Test Properties** pane. New output folder options for the test results and the Test Report allow you to select between several relative locations, which makes moving or sharing files easier, or to assign an absolute location.

For more information see Saving Test Results and Generate a Test Report.

Minor Enhancements to Simulink Element

Minor enhancements have been done in the Simulink element, including an easier way to extract time data and changing the model end time when importing all signals.

Plot Element Transition

The Vector Plot and Scalar Plot elements are being replaced by the General Plot element that was introduced in R2008b. The General Plot element supports all MATLAB plot types. You can plot any type of data, using any MATLAB plot types or Simulink data.

The General Plot element has special support for the many time-based data formats produced from a Simulink model. The supported data types include an array, a structure, a structure with time, or a time series. You can also plot Simulink data provided by test vectors, such as the Signal Builder Block test vector, the Simulink Design Verifier Data File test vector, or the Spreadsheet Data test vector.

Compatibility Considerations

The following outline describes the timing of the transition process:

- **R2009a and R2009b** — You should no longer create new Vector Plot or Scalar Plot elements. Tests containing the old elements will continue to work, but we recommend that you do not create any new ones. We recommend using the General Plot element.
- **A future release** — You can no longer create new Vector Plot or Scalar Plot elements. A warning will appear if you load a test containing the old elements. Tests containing the old elements will continue to work, but you cannot create any new ones.

  You will not lose the use of the old elements in existing tests. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element.
- **A future release** — You can no longer load tests containing Vector Plot or Scalar Plot elements. If you try to load one, you will get an error, and will have to convert the element to the General Plot element. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element.

R2009a

Version: 2.3

New Features

Bug Fixes

Compatibility Considerations

MAT-File Test Vector

A new MAT-file test vector has been added. It reads data from individual MAT-files into the SystemTest environment. For more information on this feature, see Create MAT-File Test Vectors.

Automatic Naming of Test Vectors and Test Variables

A new feature has been added to the Simulink element to make creating or mapping test vectors and test variables easier. You now have the option of automatically naming test vectors and test variables from the names used in the model inports and outports. This is particularly useful for testing Simulink models with large numbers of inports or outports.

This feature is accessed by clicking the **Mappings Assistant** button in the Simulink element. For more information about this feature, see the Desktop Help for the Simulink element (in the **Desktop Help** pane when the Simulink element is selected), the subtopics called *Using the Inport Block Mappings Assistant*, and *Using the Model Output Mappings Assistant*.

Enhanced Inport Support for the Simulink Element

Enhanced support has been added in the Simulink element for logged signals, including time series and bus signals.

You can feed data logged from a Simulink model to Inport blocks of another model using the Simulink element and a MAT-File test vector that contains the logged signal data.

A common usage scenario is to log the signals while running a model and store them in a MAT-File. Then acquire them from the Mat-File using a MAT-File test vector and map that to Inport blocks in the Simulink element. For more information on this feature, see Map Logged Signals from a Model to Inport Blocks.

Additional Editing Capability for Test Vectors

You can now edit test vectors from within elements. In prior versions of the SystemTest software, you could only edit test vectors from the **Test Vectors** pane. Now you can edit them there, as well as from within individual elements.

Plot Element Transition

The Vector Plot and Scalar Plot elements are being replaced by the new General Plot element that was introduced in R2008b. The General Plot element supports all MATLAB plot types. You can plot any type of data, using any MATLAB plot types or Simulink data.

The General Plot element has special support for the many time-based data formats produced from a Simulink model. The supported data types include an array, a structure, a structure with time, or a time series. You can also plot Simulink data provided by test vectors, such as the Signal Builder Block test vector, the Simulink Design Verifier Data File test vector, or the Spreadsheet Data test vector.

Compatibility Considerations

The following outline describes the timing of the transition process:

- **Version R2009a** — Customers should no longer create new Vector Plot or Scalar Plot elements. Tests containing the old elements will continue to work, but we recommend that you do not create any new ones. We recommend using the General Plot element.
- **Version R2009a + 1** — Customers can no longer create new Vector Plot or Scalar Plot elements. There will be a warning if you load a test containing the old elements. Tests containing the old elements will continue to work, but you cannot create any new ones.

  You will not lose the use of the old elements in existing tests. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element.
- **Version R2009a + 2** — Customers can no longer load tests containing Vector Plot or Scalar Plot elements. If you try to load one, you will get an error, and will have to convert the element to the General Plot element. Previously created Vector Plot and Scalar Plot elements will be converted to the General Plot element.

R2008b

Version: 2.2

New Features

Bug Fixes

Signal Builder Block Test Vector

If you have created a Simulink model test harness using a Signal Builder block, you can automate the running of all your test cases by integrating them into a SystemTest test. This also offers the ability to collect cumulative model coverage metrics for all your Signal Builder test cases.

For more information on this feature, see Create Signal Builder Block Test Vectors.

Multiple Test Vector Groups

The ability to create multiple test vector groups has been added to the SystemTest software. A test vector group is a collection of test vectors of the same length whose values are indexed sequentially with the other vectors in the group. Groups are commonly used for Monte Carlo and Design of Experiments (DOE) based testing, or when multiple vectors have a relationship (e.g. the mass and damping values of different throttle body types). Until R2008b, you could only create one test vector group.

For example, you could have a model of an automobile motor with two blocks – one block for the fuel system and one for the oxygen flow. Each block has a number of inports that are mapped to SystemTest vectors. In previous release you could not create a test that checks how the model behaves when performing a Monte Carlo test on both the fuel system and oxygen flow at the same time. With multiple test vector group support, you can now perform a Monte Carlo style test on both blocks, saving time in the test run.

For more information on this feature, see Create Grouped Test Vectors.

New Plot Element

The new General Plot element offers support of all MATLAB plot types. You can plot any type of data, using any MATLAB plot types, or Simulink data.

The General Plot element allows you to plot data produced from a Simulink model. The supported data types include an array, a structure, a structure with time, or a time series. Note that the element creates a line for each signal in the Simulink data. If time is not present, the signals are plotted against their indices.

You can also plot Simulink data provided by test vectors, such as the Signal Builder Block test vector, the Simulink Design Verifier Data File test vector, or the Spreadsheet Data test vector.

For more information on this feature, see **Working with the Basic Elements > Basic Elements > General Plot Element** in the SystemTest help.

Simulink Design Verifier Data File Test Vector

The Simulink Design Verifier Data File test vector can read test cases created by the Simulink Design Verifier software. In order to use this test vector, you need a Simulink Design Verifier data file with test cases.

To use this feature, you first run Simulink Design Verifier with the appropriate configuration. Then you can do one of two things:

- Generate a SystemTest harness for the model from Simulink. When it completes, a new test opens automatically in SystemTest and a Simulink Design Verifier Data File test vector is automatically created. This workflow is described in Automatically Creating a SystemTest Test Harness from Simulink Design Verifier.
- If you already have a data file from Simulink Design Verifier, you can create a test vector in SystemTest that uses the data, and configure overrides in a Simulink element. This workflow is described in Creating a Simulink Design Verifier Data File Test Vector.

Model Coverage Support for Reference Blocks

The SystemTest software now supports Model Coverage for Reference blocks. The Simulink element will allow model coverage to be gathered for referenced models.

For more information on this feature, see Simulink Model Coverage.

Run Status Hyperlink Change

Prior to R2008b, after a test ran, the **Run Status** pane contained a link “Open Test Results in Viewer.” This is no longer available. However, you can still open the results in the Test Results Viewer by selecting the **Tools > Test Results Viewer** menu or setting the automatic launch option on the **General** tab in the **Test Properties** pane.

The hyperlink in **Run Status** has been replaced by a link that opens the test results object in MATLAB.

R2008a

Version: 2.1

New Features

Bug Fixes

Compatibility Considerations

Overriding Individual Simulink Inport Blocks

You can now override individual Simulink Inport block signals via the Simulink element.

In the element, you can map test signals to individual Simulink Inport blocks using the new **Override Inport Block Signals with SystemTest Data** section on the **Mappings** tab.

For information on using this feature, see Overriding Inport Block Signals in the SystemTest documentation.

Limitations and Restrictions

When you select **Inherit from model** as the mapping for an individual Inport block, it will not support models that defined their `ExternalInput` as expressions of `t`, time series, Simulink Time Series, or TSArray.

Individual Inport mappings restrict the dimension of the mapping variable to a vector (dimension = 1).

Improvements in the Simulink Element

The Simulink element has been updated to use collapsible panes for the different mapping options, offering better usability within the element. Any mappings from previous TEST-files are loaded into the appropriate collapsible pane. The four collapsible panes consist of:

- Inport Block Signals (new feature – Importing Simulink Inports)
- Block Parameters
- To Workspace Variables
- Outports

In previous releases, when loading TEST-files, Simulink models associated with a Simulink element were not automatically loaded into memory during load time. In Release 2008a, the Simulink element will now load its model during load time, which is indicated by the load progress bar.

In previous releases, the Simulink element would not recognize To Workspace blocks stored within locked masks or block parameter fields for sample time, data type, and scaling. These options are now accessible through the element.

M API to Access Test Results in MATLAB

After you run a test that includes results you saved using the **Saved Results** in the **Test Browser**, you can view the results in MATLAB using a test results object. This feature is useful for comparing the results of separate test runs and for postprocessing of test results.

After running a test, a MATLAB variable named `stresults` is made available, allowing you to access the test results object and work with the data in MATLAB. One of the main advantages to accessing test results in the command line is that all of the MATLAB plotting tools are available to use on the results data. You can plot the data using any of the plot types MATLAB offers.

Another major use of the test results object is to quickly access and view your results in the form of a dataset object. The dataset object offers a convenient form for managing and displaying a large amount of data.

For information on using the new test results object, see Accessing Test Results from the MATLAB Command Line in the SystemTest documentation.

In previous releases, test results were not accessible during test execution. In Release 2008a, the most current test results can be accessed at any time in the test by using the `systest.testresults.getCurrent` function.

Compatibility Considerations

In previous releases, test results were managed in an undocumented MATLAB structure accessible as a variable named `stresults` or from the test results MAT-file. For Release 2008a, the test results are now managed as a documented test results object. Any dependencies on the previous undocumented test results format will need to be updated to use the new test results object.

For information on using the new test results object, see Accessing Test Results from the MATLAB Command Line in the SystemTest documentation.

Test Vector Types

In this release, three test vector types are now available, described as MATLAB Expression, Spreadsheet Data, and Probability Distribution. Test vector data for TEST-files from prior releases will be loaded as a MATLAB Expression test vector type.

Other Test Vector Changes

The Probability Distribution support now recalls the configuration values used to generate the distribution, avoiding the need to reenter the configuration values when a distribution needs to be edited.

The Probability Distribution test vector now uses default values for its fields. Previous versions configured the fields blank. You can replace the defaults with the values you want to use.

Option to Reuse Test Vector Values

You can now choose to reuse test vector values each time a test is run. This is useful for Monte Carlo testing when used in conjunction with the new Probability Distribution test vector type. To reuse test vector values, do not select the option called **Evaluate Test Vector each time the test is run** when configuring a Test Vector.

Note About This Feature

In previous releases, test vector values were always reevaluated upon loading and running the test. To maintain this behavior, in Release 2008a, any existing TEST-files will be loaded with their test vectors configured to **Evaluate Test Vector each time the test is run**, which is the new configuration option for test vectors. Any new test vectors created though will, by default, not be configured to reevaluate each time the test is run.

Reading Microsoft Excel Data into Test Vectors

A new test vector type, Spreadsheet Data, is now available to read data from Microsoft® Excel® files into the SystemTest software. You can import data from multiple files, and can read whole sheets or a subset of a sheet.

For information about using this feature, see Creating Spreadsheet Data Test Vectors in the SystemTest documentation.

Notes About This Feature

If a file has been added to a Spreadsheet Data test vector, any changes within the file (i.e., sheets being renamed, added, or removed) will not be automatically reflected within the SystemTest software. To have the SystemTest software reflect the change, remove the file from the test vector and then add it back.

Specific rows/columns from a Spreadsheet Data test vector can be selected using a new mapping option provided in the Simulink element's **Override Inport Block Signals with SystemTest Data** option on the **Mappings** tab. This selection option is not currently supported for other Simulink mapping types or other test elements.

The Spreadsheet Data test vector is officially supported on Windows® platforms and can read all Microsoft Excel spreadsheet files. This also includes CSV files. On platforms other than Windows, the SystemTest software only supports reading XLS files and does not support CSV files.

The following undo/redo actions are not currently supported for the Spreadsheet Data test vector:

- Adding, removing, moving up, moving down, selecting, or unselecting of files/ spreadsheets
- Any edits made on the **String Mappings** tab

The Spreadsheet Data test vector stores the absolute path to any files added. If the file changes location, even if the file is still on the MATLAB path, it will not be found. You should remove and add the file again if the location or name changes.

The other two test vector types (MATLAB Expression and Probability Distribution) clear their values and set their length to 0 whenever an "Evaluate" fails. For the Spreadsheet Data test vector, if an "Evaluate" fails, then the last known values, headers, and string replacements remain and will only be updated when an "Evaluate" is successful. This can lead to discrepancies between the actual values used and the way the Spreadsheet Data test vector user interface is configured.

- For example, if the test vector successfully reads Column A, the values are stored, etc. If the test vector is configured to read Columns B to C, and that leads to an error somewhere, the text fields will still show Columns B to C, even though the values that would be used are from Column A.
- In this same example, a failed evaluation would put the test vector in error. However, if you save and then reload the test vector, the error state is removed. Therefore, you could have saved this test vector reading from bad data in Columns B to C, and then reload and the red error borders are gone. When it runs (if you do not reevaluate each time the test is run, or manually, before pressing the **Run** button) it will use old

values from column A, even though the information in the **Test Vectors** pane says Columns B to C.

Test Vectors User Interface Change

The overall UI for test vectors and test variables has been redesigned into a table and embedded edit area, offering easier management and interaction. By default, the test vectors and test variables are no longer docked in the bottom left of the desktop. Instead, they are docked in the middle pane along with the **Properties** pane. A listing of test vectors and variables are also provided in the **Test Browser**.

The list of test vectors in the **Test Browser** will always reflect an alphabetized order, regardless of their order in the **Test Vectors** edit pane. The test variables tree is not currently alphabetized and will reflect the order from the edit pane.

Multiple test vectors can be selected in the edit pane to allow for multiple test vector evaluation or deletion. Multiple test vector editing, though, is not currently supported.

Test File Format Change

The following changes have been made.

Compatibility Considerations

TEST-files are being saved in a new internal binary format as of Release 2008a. Files from previous releases are compatible with Release 2008a and can be loaded without any problems. TEST-files saved in Release 2008a will not be loadable in previous SystemTest releases.

The results MAT-file generated by the SystemTest software, or those saved by the Test Results Viewer, stores data in the form of a new test results object for Release 2008a. These result MAT-files are not loadable by previous SystemTest releases.

General UI Changes

The desktop toolbar has been updated to offer a more concise set of options.

- The **Run** and **Stop** buttons have been relocated toward the right end of the toolbar.

- The buttons that launch the Test Results Viewer, Test Report, and Simulink have been removed. These options continue to be available from the menu bar.
- The Current Directory chooser at the top of the desktop has been removed. The current directory can still be changed by using the MATLAB desktop Current Directory chooser.

A progress bar has been introduced providing status when TEST-files are loaded.

The **Run Status** progress bar has been updated to an indeterminate bar, suggesting ongoing testing when working in distributed mode. The new progress bar no longer displays the percentage complete. Information on current progress is still provided via the status text above the progress bar.

The **New** button in the **Test Browser** pane, originally used to add elements, can now be used to add elements, test vectors, and test variables.

Compatibility Considerations

Three reserved names have been introduced, restricting the names that can be used for Test Vectors, Variables, Saved Results, or Derived Results (created in the Test Results Viewer). These reserved names are:

- ObsNames
- Properties
- VarNames

Any existing TEST-files that use these names will be loaded with a warning indicating what alternate name is being used in place of the restricted name.

Default Setting Changes

With the addition of a new test results object for exploring and analyzing test results, by default, the **Visualize and plot saved results by launching the Test Results Viewer** option is turned off when new TEST-files are created. The option is still available in the **Test Properties** pane.

The default desktop layout no longer displays the **Elements** pane by default. It is still available via the **Desktop** menu.

Test Results Viewer Notes

The Test Results Viewer currently supports viewing scalar, vector, time series, and image data. It is not currently compatible with any other data formats. Attempting to load other data formats will produce an error. In particular, using the new Spreadsheet Data test vector to read in multiple columns/rows of data, producing a matrix of values for each test iteration, will likely not be loadable by the Test Results Viewer. It is recommended the new test results object, which is compatible with all data formats, be used instead to view and analyze test results.

When configured to launch, the Test Results Viewer would always launch at the end of a test, even when the test was unable to run to completion or if there were no test results to view. In Release 2008a, the viewer will only launch if the test runs to completion and if test results are being saved. Otherwise the request is ignored.

R2007b

Version: 2.0

New Features

Bug Fixes

Compatibility Considerations

Distributed Testing

You can now distribute the SystemTest software tests across multiple computers or processors for faster performance. You must have a license for the Parallel Computing Toolbox™ software to use this feature.

You can set up a test and then distribute the test iterations as tasks, which are performed by different workers. No changes to the test setup are required. This can speed up the total time the test takes to execute.

Randomized Test Vectors

The SystemTest software now provides an easy way to generate randomized test vector values for your test. Probability distribution functions can be used to set up test vectors, which is useful for performing Monte Carlo analyses.

If you have the Statistics Toolbox™ software, the SystemTest software integrates with it to provide use of some of its probability distribution functions, such as `Exponential`, `Gamma`, `Lognormal`, `T`, and `Weibull`. If you don't have the Statistics Toolbox software, you have access to MATLAB probability distribution functions such as `Normal` (Gaussian) and `Uniform`.

Grouped Test Vector Support in the Test Results Viewer

The Test Results Viewer now lets you use grouped test vectors in plot configurations.

You can also show grouped test vectors as dependent variables in scatter plots. If a dependent variable is a grouped test vector, you can now explore the relationship it has with the other vectors in the group.

Run SystemTest Tests from the MATLAB Command Line

You can now run tests that you set up in the SystemTest software from the MATLAB command line. This is useful for running multiple tests as a batch or calling a test file as part of a MATLAB file.

Platforms No Longer Supported

The following platforms are no longer supported for use with the SystemTest software:

- Solaris™ 64
- PowerPC® Mac

Note that since these platforms are not supported, you cannot have any worker machines in a distributed testing cluster that use these platforms.

Invalid Characters in Element Names

The following characters are now considered invalid to include within element names:

- '
- <
- >

You cannot use these three characters in element names. If you create a new test element with one or more of these characters in the element name, then the SystemTest software throws an error dialog and the element name is reset to the default value, which is the name of the element type.

Compatibility Considerations

If you try to load an existing test with an invalid element name (containing one or more of the three characters listed above), the SystemTest software displays an error dialog indicating that the element name is invalid. The test will load successfully, but the element with an invalid name is reset to use the default name for the element.

If this occurs, simply rename the element to a name that does not contain any of the invalid characters. You rename an element by double-clicking its name in the **Test Browser**.

R2007a

Version: 1.1

New Features

Bug Fixes

Compatibility Considerations

Enhanced Limit and Tolerance Checking

The Limit Check element has new functionality that creates many more options for defining the pass/fail criteria for a test. In addition to scalar versus scalar comparisons, other comparison types are now supported: scalar versus vector, vector versus vector, and matrix versus matrix. You can now perform tolerance checking as well. This feature allows you to check absolute tolerance or relative tolerance of variables used throughout the SystemTest software.

Undo/Redo Support

Undo/redo support is now available in the SystemTest desktop.

Flexible Test Iteration Definition Using Grouped Test Vectors

Test vectors can now be grouped, in order to affect the way iterations are run. By grouping test vectors, they will be combined sequentially with the other grouped vectors, and then permuted with all the ungrouped test vectors. This gives you more control over the flow of your tests and is useful for Monte Carlo or Design of Experiments (DOE) based testing as well as defining signal groups, similar to those defined in the Simulink Signal Builder block.

Enhanced Reporting of Test Execution

Reports generated for the test execution now contain more information. For example, the new report includes test description text, snapshots of the plots generated by Plot elements, and snapshots of the models used in Simulink elements.

Test Results Viewer Save and Load Capabilities

Users can now save the plotting and analysis work done in the Test Results Viewer. Data, constraints, and plots created in the Test Results Viewer can be saved and then reloaded in order to continue working on or viewing the data, or to share it with others.

Remove Test Vectors from the Limit Check Element

Test vectors are no longer available in the **Test Variable** column of the Limit Check element since there is no rationale for that use case. The “Test Vector” choice has been removed from the drop down in the Limit Check table.

Compatibility Considerations

Any existing tests using Test Vectors will have their Limit Check tables now launch in an error state. Tests cannot have Test Vectors in the **Test Variable** column. To correct the problem, choose an available variable for the limit check or create a new variable.

R2006b

Version: 1.0.1

New Features

Bug Fixes or Enhancements

The following enhancements were made in Version 1.0.1:

- The Stop element can now be used in Pre Test. In Version 1.0, it could be used only in Main Test. It is now available in both Main Test and Pre Test.
- The Stop element now generates a hyperlink in the **Run Status** pane that gives the name of the Stop element that terminated the test and shows the element in the SystemTest desktop.

R2006a+

Version: 1.0

Introducing SystemTest

Version 1.0 was released in Web-downloadable form after R2006a, and includes the features listed below.

The SystemTest product provides a software framework for developing tests that exercise MATLAB algorithms and Simulink models. It includes predefined test elements that let you build and maintain standard test routines. You can save and share tests throughout a development project, ensuring standard and repeatable test verification from research and development through preproduction. You can map test variables into a result set for analysis.

SystemTest is integrated with MATLAB, enabling you to use the MATLAB scripting language and analysis tools to explore large data sets and gain further insight into the behavior of the system. Key features include:

- Develops, manages, and edits test structures
- Uses predefined test elements to test MATLAB algorithms and Simulink models
- Incorporates MATLAB scripts directly into a test
- Maps MATLAB and Simulink workspace variables to test variables
- Offers reusability, so that after you design a test, you can save it for later use by you or others
- Visualizes and analyzes multidimensional test results