

# The Reactis® Communicator

**To** : Reactis Users  
**From** : Reactive Systems, Inc.  
<http://www.reactive-systems.com/>  
**Date** : January 25, 2006

## IN THIS INSTALLMENT:

1. Reactis V2006 Now Available.
2. RSI Logs Strong Growth in 2005.
3. Model-Based Testing with Reactis at Azure Dynamics.
4. Revamped Reactis FAQs Now Available.
5. Jump-Start Reactis Deployment with On-Site Training.

## 1 Reactis V2006 Now Available

We are pleased to announce the availability of Reactis V2006, which includes numerous new features and enhancements since the previous major release in May of 2005. Some highlights are below. For more details please see the revision history in the Reactis User's Guide which is included in the distribution and is also available from: <http://www.reactive-systems.com/products.msp>

The following are some of the features added since V2005.

### 1.1 New Test Suite File Format

The V2006 Reactis release introduces a new file format for test suites (.rst files). The new format allows Reactis to access a test suite directly on disk instead of requiring the entire suite to be loaded in memory. The format enables Reactis to manipulate dramatically larger test suites. The new capability is especially useful when importing very large test suites created outside of Reactis. The new format also enables several other new features including the following:

- Tests may now have user-specified names.
- Detailed logs for each test and the suite as a whole are maintained by Reactis.

When loading a test suite stored in the pre-V2006 format, the test suite will be automatically converted to the new format. Test suites can also be exported to the pre-V2006 format.

## 1.2 New Tools for Covering Timer Conditions

Creating tests that cause interruptible timer conditions to reach their upper bound is often a challenge. An interruptible timer condition is a model variable that changes its value from a start value to an end value by some increment. In such cases, it is often possible for the variable to be reset to its start value as the system executes. V2006 includes several new features to assist you in creating tests that cause timer conditions to reach their upper bound.

## 1.3 Enhanced Model Navigation

1. Dragging with left mouse button in the main window will now scroll the model.
2. When hovering over an item in the hierarchy panel that is a child of the system currently displayed in the main panel, Reactis now highlights the label of the tree item and the child in the main window.
3. Added a “Go to Parent” button to the Reactis toolbar.

## 1.4 License Manager Enhancements

1. The Reactis License Manager may now be configured to log license usage to two different locations: the Windows Application Event Log and a log file. The logs include information about when licenses are granted and released as well as who occupies them.
2. In the “User Info” tab of the “Settings” dialog, an option is now available to automatically set the name field to the Windows username of the person currently logged in.

## 1.5 API Improvements

1. The API now uses Matlab version as set in the Reactis GUI.
2. Added file “libreactis.lcc.lib” to API. This enables applications using the Reactis API to be compiled using the default compiler distributed with Matlab.

## 1.6 Newly-Supported Simulink / Stateflow Features

V2006 adds support for many new Simulink blocks and features. Some of these are listed below.

1. Support for Look-Up-Table Dynamic blocks.
2. Support Simulink Signal Viewer Scopes.
3. Support Simulink “From Workspace”, “From File” and “Signal Builder” blocks.
4. Support for exporting Stateflow graphical functions from a library.
5. Support models created with Matlab R14.3. Not all new features of R14.3 are supported yet.
6. Support for alias types created with Simulink.NumericType or Simulink.AliasType. These alias types are now supported both in Simulink and Stateflow. Alias types are dereferenced and show up in Reactis as the native Simulink base types.

## 1.7 Other Enhancements

1. Configuration variables of type array and structure are now supported.
2. Reactis now flags a number of different types of modeling constructs that can lead to unexpected behavior.
3. If an error occurs during test generation, Tester will add the current test (which caused the error) to the test suite before terminating.
4. Added a “Forward” button to the Tester launch dialog that allows the user to return to the Tester results dialog after clicking “Back”.
5. Allow import of test suites (.rst files) that do not exactly match the current model. This can be achieved by selecting the **Test Suite -> Import** menu item and then selecting a test suite file. Types will be adapted and ports and configuration variables added or removed to fit the current model.
6. In user guided simulation mode, provide a menu to switch all inputs to “random” or “user guided” mode at once.
7. Show current step number when in user-guided or random simulation mode.
8. Improved the type editor dialog when editing array types. The dialog now handles 2-dimensional arrays and includes a capability to copy and paste type-specifications for the elements of the array.
9. Track branch coverage for Relay blocks.
10. In Simulator, when one or more tests are selected to be run, the “number of steps” entry box in the toolbar will be pre-filled with the number of steps selected.
11. When choosing configuration variables, the list of workspace data items is now sorted alphabetically.
12. Tester now obtains better boundary value coverage for vector inputs.

## 2 RSI Logs Strong Growth in 2005

We are pleased to report that 2005 was a good year for Reactis. The number of companies using Reactis almost doubled and, due to strong growth within existing customers, active licenses more than doubled. Use of Reactis increased in both the automotive and aerospace sectors.

## 3 Model-Based Testing with Reactis at Azure Dynamics

A recent article published on The MathWorks website describes how a Model-Based Design process that includes Reactis helped Azure Dynamics achieve a level of test coverage “reserved for highly regulated industries.” The Azure team uses Reactis in the development of hybrid-electric and electric powertrain applications. The article is available at:

<http://www.mathworks.com/industries/auto/userstories.html?file=11337>

## 4 Revamped Reactis FAQs Now Available

A new version of the Reactis FAQs is now available from the RSI website. It has been reorganized for easier navigation and includes numerous new questions and answers. Please keep the questions coming... The FAQs may be found at:

<http://www.reactive-systems.com/faq.msp>

## 5 Jump-Start Reactis Deployment with On-Site Training

Reactive Systems now offers one- and two-day Reactis training courses delivered by a qualified instructor at your site. The hands-on courses include numerous exercises to introduce engineers to model-based testing and validation with Reactis. No prior experience with Reactis is required. For more information on the cost-effective way to spread the benefits of Reactis within your organization, please send email to [info@reactive-systems.com](mailto:info@reactive-systems.com) or call 703-534-6458.

Best Regards,  
The Reactis Team

This is installment sixteen of “The Reactis Communicator”, a low volume mailing list for conveying information about Reactis, RSI’s embedded software design automation tool suite. Reactis enables users to deploy model-based software testing and validation to dramatically reduce the costs of testing embedded control software. The tools are designed for use in conjunction with the Simulink and Stateflow modeling and simulation environments offered by The MathWorks, Inc.

If you are no longer interested in receiving information about Reactis, please see the instructions below for removing yourself from the list and we apologize for the intrusion.

*Reactis is a trademark of Reactive Systems, Inc. MATLAB, Simulink, and Stateflow are registered trademarks of The MathWorks, Inc.*