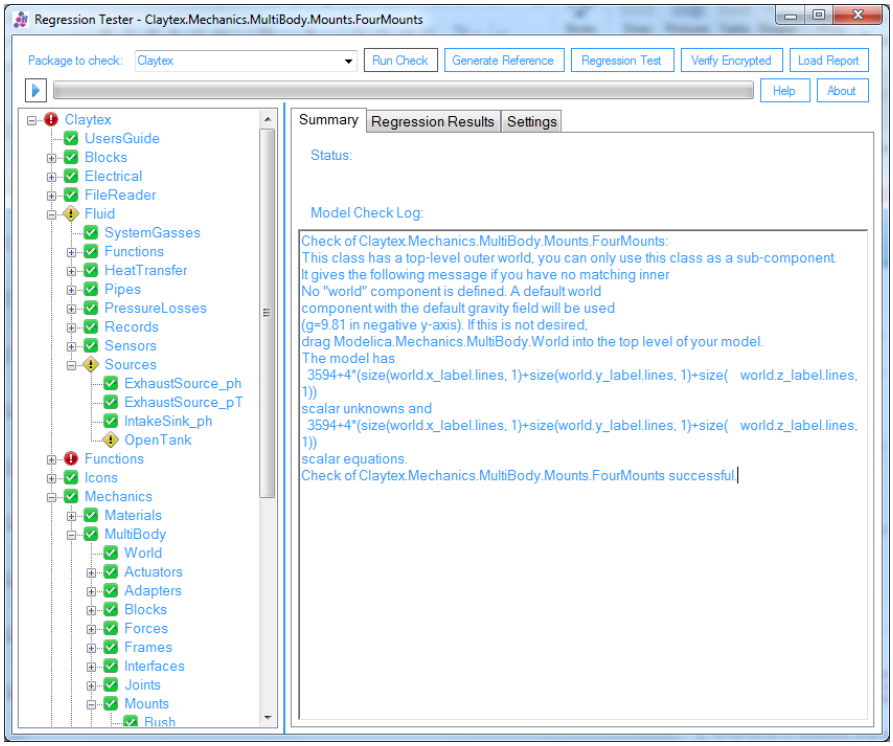
RegressionTest Tool for Dymola

Key Features



- Standalone tool that controls Dymola to run library tests
- Automatically check every model in a Modelica library
- Generate reference results from experiments in a Modelica library
- Perform a regression test to compare a new version of a library to a reference set of results
- Verify that an encrypted library produces the same results by comparing the results to a reference set
- Present the results in an easy to browse format
- Supports command line operation with XML report generation
- Supports Windows 64-bit

Overview

The RegressionTest Tool for Dymola has been developed for library developers to help them improve the quality of library releases. It provides automated routines for checking models, generating reference results and running regression tests. It is a standalone application that controls Dymola and call functions from within a supplied Modelica library.

The tool can be run interactively using the Graphical User Interface shown above or it can be run from the command line. Using the command line option enables tests to be scheduled to run nightly in a fully automated manner. Each test that is run by the tool generates an XML report that can be loaded in to the GUI for analysis.

The test results are presented in a hierarchical view that recreates the library structure with different icons used to highlight whether a model is correct, generates warnings or fails the test. When there are warnings or models fail the tests within a package these are propagated up through the library structure so that they can be quickly identified and located within the library.

Model Check

Every class in the selected library is checked using Dymola and the results summarised to quickly identify models that aren't correct and those that generate warnings.

An example of running this test is shown on the previous page. When you click on the name of any class in the library the translation log generated by Dymola is displayed so that warnings and errors can be quickly understood and models corrected.

When running a check, the tool is able to identify functions that are marked as user interactive functions or valid templates and adjust how the class is checked in an appropriate way. This helps avoid false error messages.

Regression Test

A regression test involves the simulation of a new version of a model and comparing the translation log and results from this new simulation against a reference result set. The results of a Regression Test are a comparison of the translation logs and a simulation results.

The translation logs are compared to identify changes in the size of the equation systems present in the model. This is shown in the top figure on the right.

The comparison of the simulation results is done using the state variables and a user-defined list of variables that can be stored with the model. The variable values are compared at each time step and any difference that is outside a specified tolerance is identified as a failure of the regression test. The regression test results can be viewed as shown in the bottom figure on the right.

Verify Encrypted

A regression test can also be performed using the encrypted version of a library. In this case the reference results are used to determine the names of the experiments to be used. The comparison of the translation logs and simulation results is carried out in exactly the same way that a normal regression test is run.

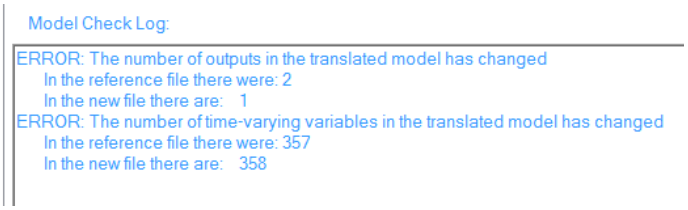
This enables libraries to be checked after encryption to ensure that the experiments work as intended.

Generate Reference

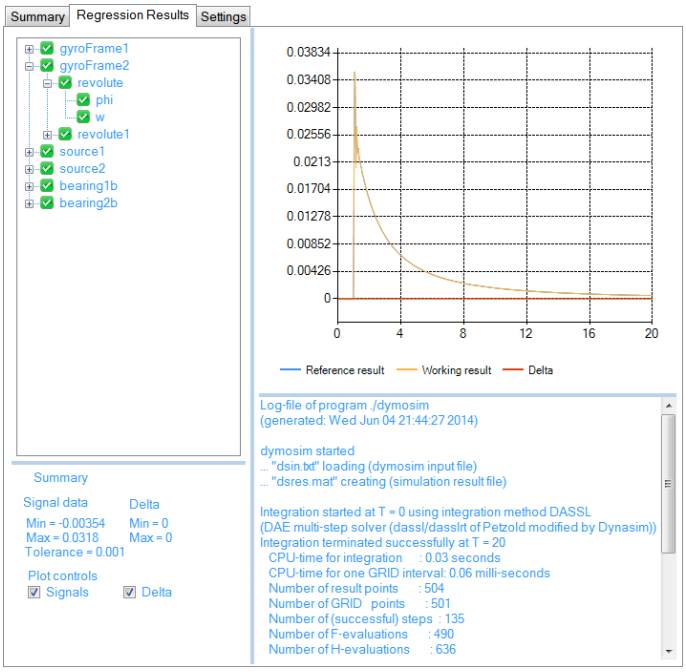
The selected library is scanned for models that are marked as experiments (i.e. tests cases) and these are then simulated using the settings stored in the model to generate a reference result file.

An experiment is identified by searching for the Modelica annotations that store the simulation settings within the model code. There are some additional annotations available that are specific to this tool that can be used to include other settings that are not part of the standardised annotations. For example, these could be used to store the Dymola specific real-time simulation settings that must be applied to run the model correctly.

When this test is run the simulation result file and the translation log file are stored using a fixed naming structure for the library directory which includes the version number and build number details.



Comparison of translation statistics



Comparison of simulation results

Claytex Services Limited, Edmund House, Rugby Road, Leamington Spa, CV32 6EL, UK
Tel: +44 1926 885900 – Fax: +44 1926 885910 – Web: http://www.claytex.com - Email: info@claytex.com