Functional testing for critical software with RapiTest

Rapi**Test**

Rapita Systems' on-target test solution, Rapi**Test**, drives the inefficiencies out of functional testing in critical software verification projects. By automating the creation and execution of unit, integration and system tests, it reduces the cost of software verification, particularly in the avionics industry.

Testing with RapiTest

Rapi**Test** uses input test suites to automate functional testing by building a test harness, executing tests, collecting data and generating reports.

It accepts a variety of test input formats supplied through a simple user interface. By integrating with your existing build system, Rapi**Test** reduces the overheads needed to get your test project started.

Rapi**Test** supports large testing projects by offering a portable test environment and retaining references to code and its revision in test reports.

Key features

- Automated platform for unit and system testing both **on-target** and **on-host**. Integrate testing into your existing build process to improve the efficiency of your testing.

- **Efficient test generation**. Minimize build cycles and test rig utilization by running multiple tests per build.

- **Fast incremental builds**. Rapi**Test** builds only critical files when your code base changes, even in systems with very large code bases.

- **Minimize "usercode"** by writing tests in the powerful spreadsheet and script formats of Rapi**Test**. These let you write test cases for generics or multi-dimensional tables without "usercode".

- **Portable test environments**. Share test environments between multiple testers, even where source code is unavailable to them.

- **Advanced support for Ada**. Rapi**Test** handles even the most complex Ada code constructs such as generics, nested functions and private types.

- **Flexible stubbing options**. Write both simple and complex behavior for stubbed functions easily. Rapi**Test** supports all types of stubbing behavior, including stubs, fakes, mocks, spies and dummies.

- **Interface with other RVS plug-ins**. Integrate coverage, timing and scheduling analysis into your testing process.

Automation of software testing

RapiTest makes it easy to write, run and report on software tests by providing an automated workflow that can be managed through a simple user interface or the command line:

- Generate test templates quickly

- Automate on-target and on-host testing with fast incremental builds and efficient test generation

- Highlight failed tests to review, rewrite and rerun

- Merge reports from different test runs and builds

Powerful test formats

A testing tool is only as powerful as its inputs. The test authoring technology behind RapiTest makes it quick and easy to write tests:

- Minimize usercode by defining even complex tests easily using spreadsheet or script formats

- Write a variety of stub behaviors without usercode

Tests					
Test details	Text	Call or stub	Variable	Operation	Value
New_Test					
#test valid integer values					
Name					From zero
Requirements					11
		Subtract_One	Value	set	0
		Add_One	Value	check	:
		:stub	return	set	:
		:uut	return	check	-1
End_Test					

Easy to use, easy to manage

RapiTest is designed for ease of use, even in large scale projects including multiple test engineers and sites. It provides support through the following features:

- Analyze source code only once per project to reduce unnecessary overheads

- Portable test environments let you use a single project file throughout a team

- Add files such as requirements to projects for better traceability

- A variety of results filtering and search options to highlight project progress over time

- Easy to follow tutorials and detailed documentation

Get ready for certification

RapiTest is built with safety in mind. Its DO-178B/C qualification kits and qualified target integration kits help reduce the cost of certification.

Integration with build systems

RapiTest is flexible and fits directly into existing build systems through a range of strategies:

- Using a compiler wrapper to intercept calls between the build system and the compiler

- Using a clone of the project to automatically build the instrumented software

- Adapting the build system to insert direct calls to RVS tools

Integrated testing environment

As part of the Rapita Verification Suite, RapiTest offers a gateway to other verification solutions:

- RapiCover for structural coverage analysis up to and including MC/DC

- RapiTime for detailed timing analysis including high water mark and worst-case execution time

- RapiTask for scheduling analysis in multi-thread and multi-core systems

Language support

- Ada 83-2012, support for GNAT Pro™ and Green Hills™ compilers among others

  - Handles even the most complex code constructs such as generics, nested functions and private types

- Support for C and agreed C++ language features

Licensing

- Enterprise license provides a single price for all features, transferrable across projects

- One-year support and maintenance included in purchase price