

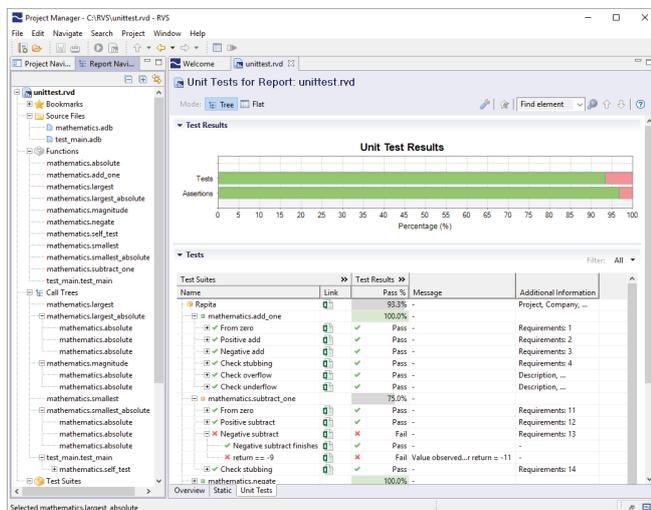
Functional testing for critical software with RapiTest

RapiTest

How can RapiTest help you?

RapiTest drives the inefficiencies out of functional testing in critical software verification projects. By automating the creation and execution of unit, integration and system tests from input tests written in easy to use test formats, it reduces the cost of critical software verification.

RapiTest's test formats make it easy to write tests at any level (unit, integration, system) and define all types of stub behavior used in the industry.



Benefits of using RapiTest

- 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**. RapiTest 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 RapiTest. 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**. RapiTest 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. RapiTest 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.

How does RapiTest work?

RapiTest 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 development environment, RapiTest reduces the overheads needed to test your code.

RapiTest supports large testing projects by offering a portable test environment and retaining references to code and revision numbers in test reports.

Contexts						
Context name	Shorthand	Parameter	Type			
Mathematics						
Scope						
Subprogram name	Shorthand	Parameter	Type			
Absolute		Value	Integer			
		return	Integer			
Stubs						
Subprogram name	Shorthand	Parameter	Type	Default		
# No stubs defined for these tests						
Tests						
Test details	Meaning	Call or stub	Variable	Operation	Value	Value
New_Test						
Name					Zero	Positive
Requirements					1	2
		Absolute	Value	set	0	5
			return	check	0	5
End_Test						

Key features

Functional testing

- 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
- 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
- Results filtering and search options to highlight project progress over time

Test authoring

- Minimize usercode by defining even complex tests easily using spreadsheet or script formats
- Write a variety of stub behaviors without usercode
- Tutorials, documentation and training to get started easily

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

Integration with build system

- Multiple strategies available:
 - Compiler wrappers
 - Clone integration
 - Scripting into build system directly
- Support for very large code bases
- Shared integration with other RVS tools

Target integration

- Support for data collection using CAN, Serial, Ethernet, debuggers and our own **RTBx** data logger
- Extremely low overhead map data collection can be configured with a single assembly instruction
- No library/run-time dependencies or dynamic memory requirements

Tool qualification

- Qualification kit and service to support DO-178B/C and ISO 26262 tool qualification

Third party integration

- Tools such as Mx-Suite™, MATLAB Simulink and GNAT GPS
- Continuous build servers e.g. Jenkins, Bamboo
- Debuggers e.g. Lauterbach, iSYSTEM

GUI

- Summary and detailed views
- Treemaps to view test results of your system at a glance
- Code viewer:
 - View source code alongside pre-processed and instrumented code
 - Color-coded by whether tests passed or failed
- Compare reports
- Database-like search function

Licensing

- Enterprise license gives you access to new versions, support and maintenance
- One-year support and maintenance included in purchase price
- Single price for all features
- Licenses transferrable across projects



Rapita Systems Inc.
41131 Vincenti Ct.
Novi, MI 48375

Tel (USA):
+1 248-957-9801

Rapita Systems Ltd.
Atlas House, Osbaldwick Link Road
York, YO10 3JB
Registered in England & Wales: 5011090

Tel (UK/International):
+44 (0)1904 413945