

# **Product brief**

# Functional testing for critical software with RapiTest

# RapiTest

### How can RapiTest help you?

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 from input tests written in easy to use test formats, it reduces the cost of critical software verification.

Rapi**Test**'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.

Project Manager - C:\RVS\unittest.rvd - RV	S				-			
Edit Navigate Search Project Win	idow Help							
🖌 👝 👘 👘 🗘 🗸 🔶	•							
Project Navi 🐮 Report Navi 🖤 🗆	Welcome In unittest.rvd 🖾							
E 🗄 🕏								
munittest.rvd ^	🔄 Unit Tests for Report: unit	test.rvo	d					
Bookmarks	Mode: Tree 🗔 Flat	AL 24	Find element	0.017				
E Source Files	model te free till Flat	Ø. 1. Ma		0.010				
mathematics.adb	* Test Results							
Institutionado								
E G Functions								
mathematics absolute								
mathematics add one		1 1	1 1 1			1		
mathematics.largest	Tests							
mathematics.largest absolute	Assertions -							
mathematics.magnitude								
mathematics.negate	0 5 10 15 20	25 30	35 40 45	50 55 60 65 70 rcentage (%)	75 80 85 90	95 100		
mathematics.self test			Pe	rcentage (%)				
mathematics.smallest								
mathematics.smallest absolute	▼ Tests					o All Y		
mathematics.subtract_one						AUL -		
test_main.test_main	Test Suites	*	Test Results 🏼			^		
🕀 🗄 Call Trees	Name	Link	Pass %	Message	Additional Information			
mathematics.largest	Rapita	0	93.3%		Project, Company,			
mathematics.largest_absolute	mathematics.add_one		100.0%	•				
mathematics.absolute	- E 🗸 From zero	0	<ul> <li>Pass</li> </ul>		Requirements: 1			
mathematics.absolute	— III  V Positive add	0	<ul> <li>Pass</li> </ul>	-	Requirements: 2			
mathematics.absolute	—	۵	<ul> <li>Pass</li> </ul>		Requirements: 3			
mathematics.magnitude	—	0	<ul> <li>Pass</li> </ul>		Requirements: 4			
mathematics.absolute		0	<ul> <li>Pass</li> </ul>		Description,			
mathematics-absolute		0	<ul> <li>Pass</li> </ul>		Description,	_		
mathematics.smallest	mathematics.subtract_one		75.0%					
mathematics.smallest_absolute		۵)	<ul> <li>Pass</li> </ul>		Requirements: 11			
mathematics.absolute	─	۵)	<ul> <li>Pass</li> </ul>		Requirements: 12			
mathematics.absolute	· · · · · · · · · · · · · · · · ·	<b>Q</b> 1	× Fail		Requirements: 13			
mathematics.absolute	<ul> <li>Negative subtract finishes</li> </ul>		<ul> <li>Pass</li> </ul>					
E test_main.test_main	× return == -9	0		Value observedr return = -11				
		0	<ul> <li>Pass</li> </ul>		Requirements: 14			
It mathematics.self_test			100.0%					
Test Suites	mathematics.negate     Overview Static Unit Tests		100.0%					

# **Benefits of using RapiTest**

- Automated platform for unit and system testing both ontarget 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?

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 development environment, Rapi**Test** reduces the overheads needed to test your code.

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

ontexts						
Context name Mathematics	Shorthand	Parameter	Туре			
соре						
Subprogram nan Absolute	néShorthand	Parameter Value return	Type Integer Integer			
tubs						
Subprogram nan # No stubs defin			Туре	Default		
ests						
Test details	Meaning	Call or stub	Variable	Operation	Value	Value
New_Test						
Nam	e				Zero	Positive
Requirement	s				1	2
		Absolute	Value return	set 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
- Automatically reformat spreadsheet tests to ensure readability and easy review
- 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<sup>™</sup> and Green Hills<sup>™</sup> 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 **RTB**x 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

- SCM systems such as Apache<sup>®</sup> Subversion<sup>®</sup>
- Tools such as Mx-Suite<sup>™</sup>, MATLAB<sup>®</sup> Simulink<sup>®</sup> and GNAT GPS
- Continuous build servers *e.g.* Jenkins, Bamboo
- Debuggers e.g. Lauterbach, iSYSTEM

#### Integrated testing environment

- Summary and detailed results 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
- Link tests to requirements to view status of tests for each requirement
- 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

Email: enquiries@rapitasystems.com | Website: www.rapitasystems.com Document ID: MC-PB-104 RapiTest v9