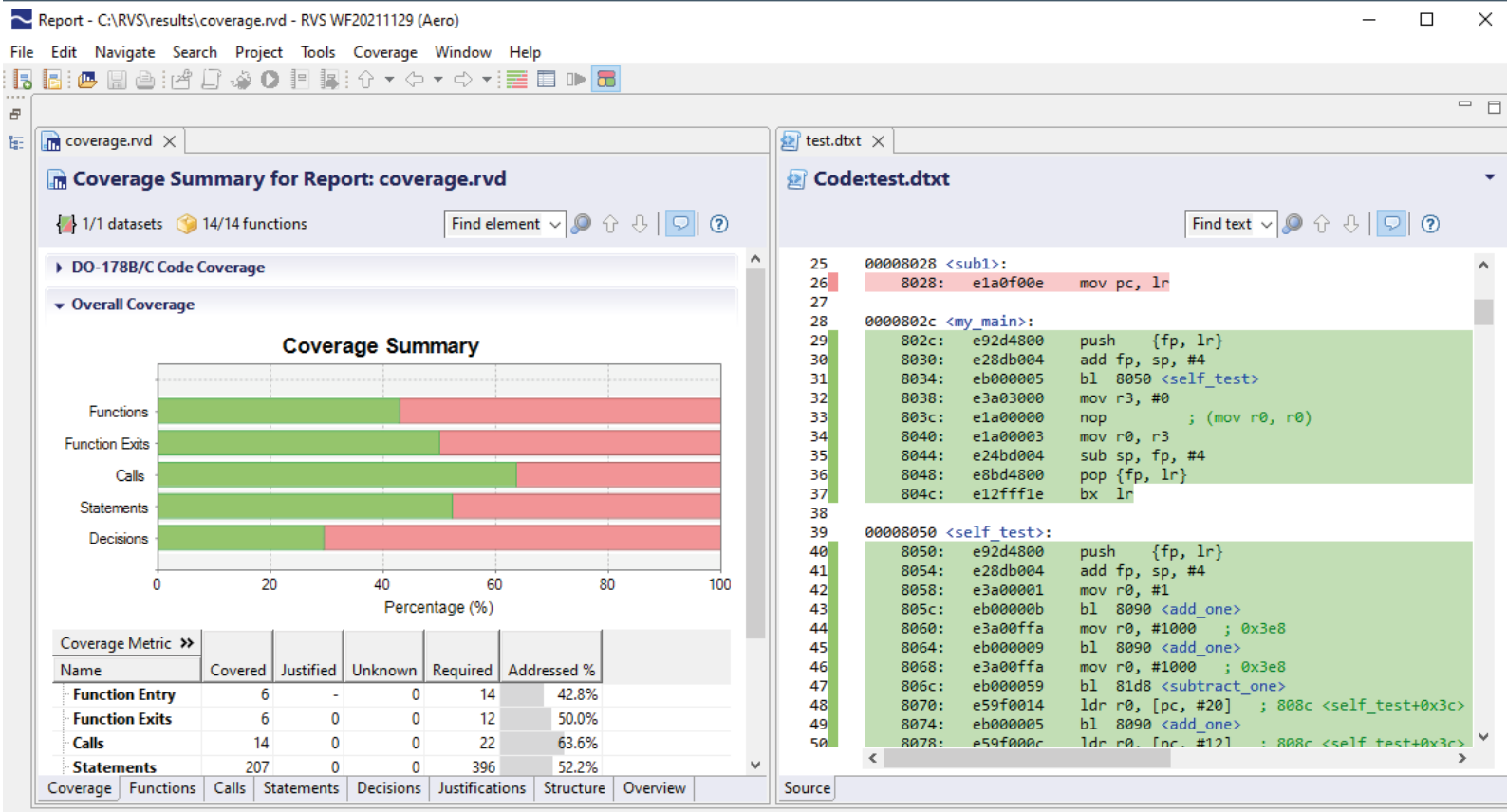


RAPITA SYSTEMS
A DANLAW COMPANY

Safety through quality

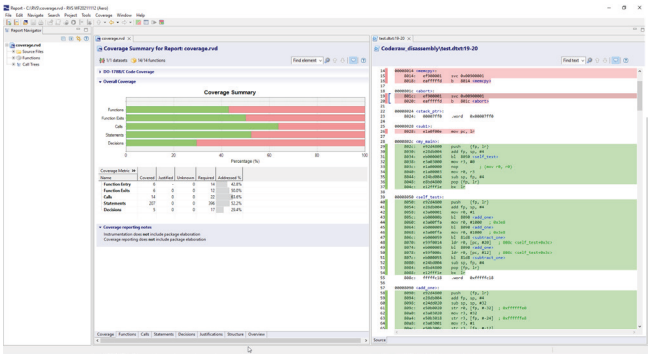
PRODUCT BRIEF

Zero-footprint coverage analysis with Rapi**Cover***Zero*

Product brief: Rapi**Cover**^{Zero}

How can Rapi**Cover**^{Zero} help you?

Rapi**Cover**^{Zero} lets you observe the structural coverage achieved during the execution of object code from critical software without needing to make any modifications to, or even have access to, your project's source code.



*Coverage results collected from Rapi**Cover**^{Zero} alongside assembly code*

Benefits

Verify the structural coverage achieved from tests of critical software without needing:

- Any instrumentation.
- Project source code.
- Any modification to your development environment.

Rapi**Cover**^{Zero} use cases

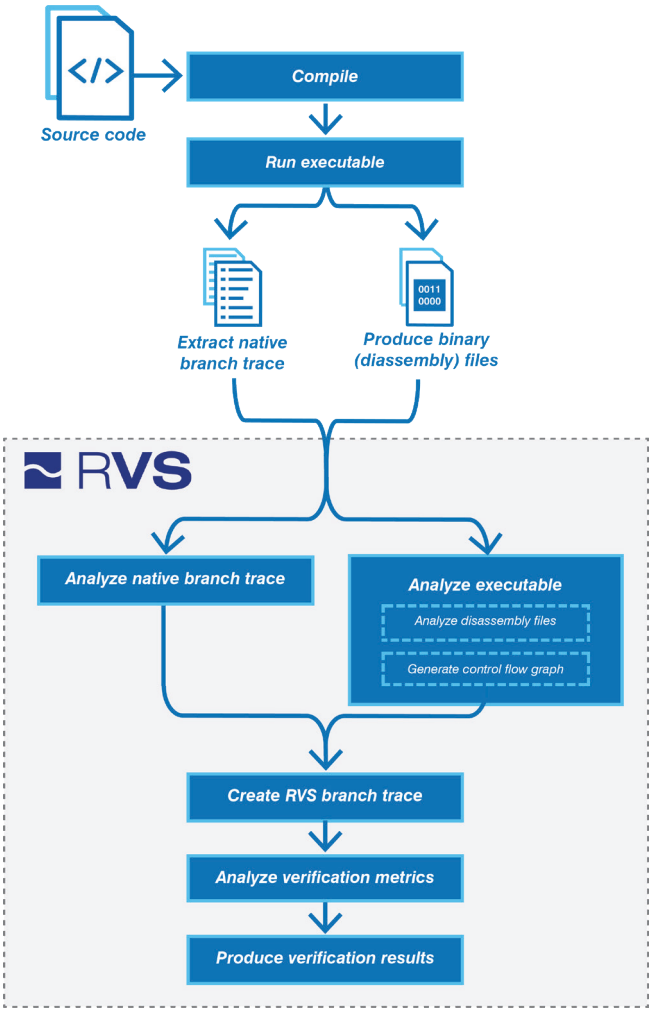
- Structural coverage analysis for third-party libraries.
- Structural coverage analysis with no impact on the code base or development environment.
- View the mapping between source code and object code.
- Address avionics software guidelines for structural coverage analysis: DO-178C, ED-12C, MIL-HDBK-512C, AA-22-01, AMACC...
- Address space software standards for code coverage analysis: NASA NPR 7150.2d and ECSS-E-ST-40C.
- Address ISO 26262 requirements for code coverage analysis.

How does Rapi**Cover**^{Zero} work?

Rapi**Cover**^{Zero} reconstructs information on software execution behavior by matching branch trace information collected from the hardware (which must support this) with a control flow graph produced from a disassembly of the software binary.

Having matched this data, a reconstructed branch trace is created, which can be used to analyze the coverage of the executable code that was achieved while it ran.

The branch trace is a crucial component of the analysis process and this must be available in the existing development environment through the CPU and/or external hardware being used.



*Rapi**Cover**^{Zero} verification process*

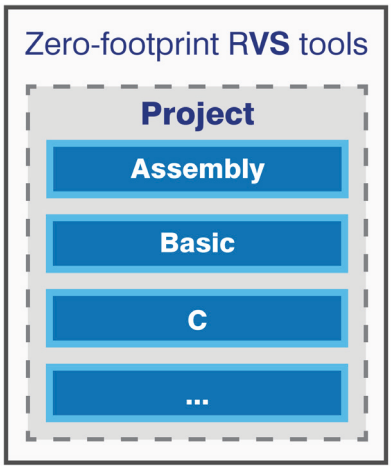
Key features

Structural coverage analysis

- Statement, function, decision and branch coverage directly from object code
- Merge coverage from different test runs, builds and strategies
- Optimized analysis profiles for avionics, space and automotive standards and guidelines

Language support

- Any language that targets machine code
- Mixed source languages



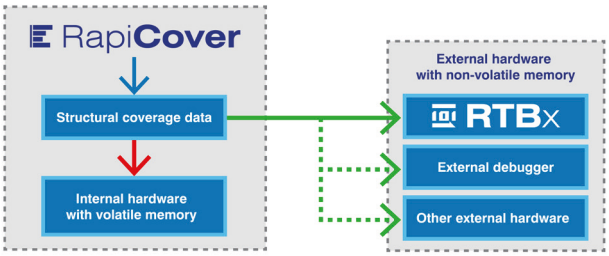
*Zero footprint R**VS** tools can be used to analyze software in any language that targets machine code*

Supported platforms

- It must be possible to produce and collect branch trace or branch map information from the platform during program execution
- It must be possible to observe context switch information from executables on the platform
- Platform Support Package required to interface between Rapi**Cover**^{Zero} and platform (see *Platform Support Packages*)
- To assess whether a Platform Support Package is available for your platform, see the compatibility tab on our Rapi**Cover**^{Zero} product page
- We can develop additional Platform Support Packages to support Rapi**Cover**^{Zero} analysis for compatible platforms

Integration support

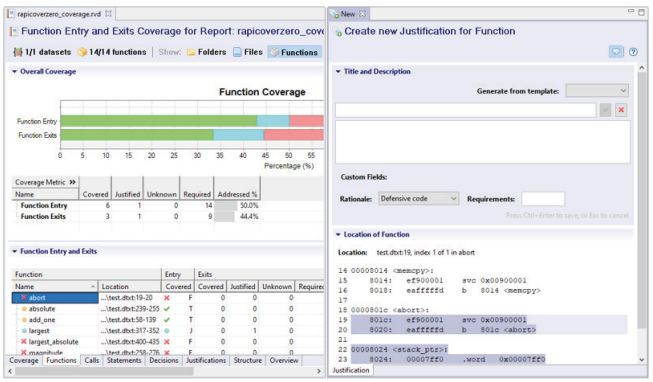
- Automatable testing environment
- Support for very large code bases
- No library/run-time dependencies or dynamic memory requirements
- Coverage analysis across power cycles (subject to hardware requirements)
- Shared integration with other zero-footprint R**VS** tools
- Continuous build servers e.g. Jenkins®, Atlassian Bamboo®
- Multicore support (depending on hardware support)



*Analyze structural coverage across power cycles with Rapi**Cover**^{Zero}.*

Justifications

- Assign justifications to manually mark code as covered by analysis
- Apply custom fields and templates to justifications
- Apply justifications to multiple locations
- Migrate justifications when code changes
  - Smart technology identifies new locations for justifications for review
- Import justifications from and export to third-party tools
- Multi-user editing support



Apply justifications to mark code as covered by manual analysis

Integrated testing environment

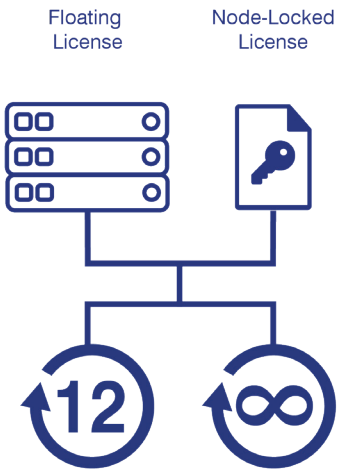
- Summary and detailed results views
- Project and code base insights including code complexity, treemaps, call dependencies, and LOC
- Filter results by subprogram
- Code viewer:
  - View object code alongside source code, where available
  - Color-coded by analysis type and whether code is covered, uncovered or justified
  - View the mapping between object code and source code
  - View missing coverage
- Compare reports
- Database-like search function
- Multiple export formats e.g. text, XML, CSV, HTML
- Multi-user testing environment

Compatibility

- Runs on host operating systems
  - Windows® 10+ and Windows Server® 2019+
  - Linux® distributions including Ubuntu® and Red Hat®
- Results can be collected from systems without supported operating systems and transferred to a supported system for analysis

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 transferable across projects



*Rapi**Cover**Zero has flexible licensing options, letting you get the most from the software depending on your needs.*

Should I use Rapi**Cover** or Rapi**Cover**Zero?

Rapi**Cover**Zero offers many benefits, but in some cases Rapi**Cover** may be more appropriate for you. Consult Table 1 below to decide if Rapi**Cover**Zero or Rapi**Cover** is best for you. For more information, contact us at info@rapitasystems.com.

Table 1. Comparison of key Rapi**Cover** and Rapi**Cover**Zero features

Feature	**RapiCover**	**RapiCoverZero**
Works without source code	No	Yes
Works without Instrumentation	No	Yes
Integration with development environment	Integration needed	No integration needed
MC/DC analysis	Yes	No
Tool qualification support	Yes	In development - contact us for more information
Trace size and data processing time	Depends on applied instrumentation	Typically larger trace and longer data processing times
Supported platforms (target, data collection mechanism)	Flexible, almost any platform supported	Requirements on platform (branch trace and context switch information must be available), PSP needed

Platform Support Packages

To enable Rapi**Cover**^{Zero} analysis on a specific platform, Platform Support Packages (PSPs) are needed for Rapi**Cover**^{Zero} to interface with that platform in order to do the following:

- Convert the specific format of native branch traces generated by the platform into a format that Rapi**Cover**^{Zero} understands and can use for subsequent analysis.
- Disassemble the object code to understand the structure and control flow of the code so this can be used for subsequent Rapi**Cover**^{Zero} analysis.

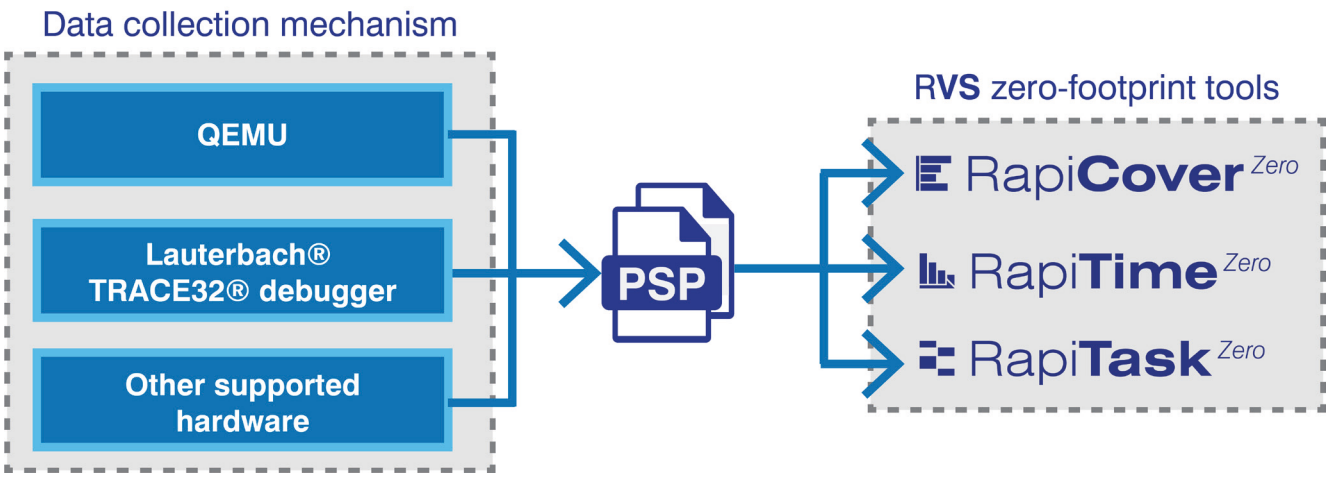
Each PSP is designed to support various components of a platform. These include:

- The compiler(s) used to generate executables to be analyzed by Rapi**Cover**^{Zero}
- The instruction set of object code to be analyzed by Rapi**Cover**^{Zero}
- The native branch trace format generated from the platform – this depends on the mechanism used to collect branch trace* information, which may be the target hardware (or simulator) or a third-party device e.g. debugger.
- The real-time operating system on which executables to be analyzed by Rapi**Cover**^{Zero} are to be run.

** In some circumstances, alternate data collection strategies such as using a branch map may be possible.*

Different PSPs are needed to support analysis by Rapi**Cover**^{Zero} when any of the above items are different between two platforms. PSPs that support Rapi**Cover**^{Zero} analysis also support analysis by Rapi**Time**^{Zero} and Rapi**Task**^{Zero}. For more information on how Zero-footprint PSPs support analysis by zero-footprint R**VS** tools including Rapi**Cover**^{Zero}, see our Requirements for zero-footprint RVS analysis Technical note.

To see whether we have already developed PSPs compatible with the components on your platform, see the compatibility tab on our Rapi**Cover**^{Zero} product page. If we have not yet developed PSPs compatible with one or more components of your platform, we may be able to develop them. For more information, contact us at info@rapitasystems.com.



*A Platform Support Package (PSP) is needed for R**VS** to interface with the development environment it is used in*

RAPITA Systems
A DANLAW Company

About Rapita

Rapita Systems provides on-target software verification tools and services globally to the embedded aerospace and automotive electronics industries.

Our solutions help to increase software quality, deliver evidence to meet safety and certification objectives and reduce costs.

Find out more

A range of free high-quality materials are available at:
rapitasystems.com/downloads

SUPPORTING CUSTOMERS WITH:

Tools	Engineering Services	Multicore verification
Rapita **Verification Suite**:	V&V Services	**MACH**[178]
Rapi**Test**	Integration Services	Multicore Timing Solution
Rapi**Cover**	Qualification	
Rapi**Time**	SW/HW Engineering	
Rapi**Task**	Compiler Verification	

Contact

Rapita Systems Ltd.
Atlas House
York, YO10 3JB
UK
+44 (0)1904 413945

Rapita Systems, Inc.
41131 Vincenti Ct.
Novi, Mi, 48375
USA
+1 248-957-9801

Rapita Systems S.L.
Parc UPC, Edificio K2M
c/ Jordi Girona, 1-3
Barcelona 08034
Spain
+34 93 351 02 05

rapitasystems.com

linkedin.com/company/rapita-systems

info@rapitasystems.com