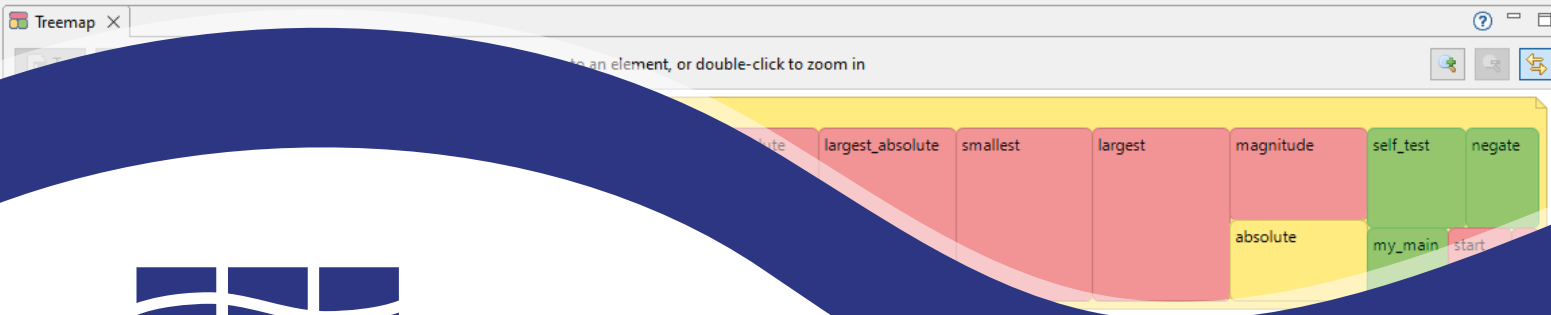


### Code:test.dtx

```

25 00008028 <sub1>:
26 8028: e1a0f00e mov pc, lr
27
28 0000802c <my_main>:
29 802c: e92d4800 push {fp, lr}
30 8030: e28db004 add fp, sp, #4
31 8034: eb000005 bl 8050 <self_test>
32 8038: e3a03000 mov r3, #0
33 803c: e1a00000 nop ; (mov r0, r0)
34 8040: e1a00003 mov r0, r3
35 8044: e24bd004 sub sp, fp, #4
36 8048: e8bd4800 pop {fp, lr}
37 804c: e12fff1e bx lr
38
39 00008050 <self_test>:
40 8050: e92d4800 push {fp, lr}
41 8054: e28db004 add fp, sp, #4
42 8058: e3a00001 mov r0, #1
43 805c: eb00000b bl 8090 <add_one>
44 8060: e3a00ffa mov r0, #1000 ; 0x3e8
45 8064: eb000009 bl 8090 <add_one>
46 8068: e3a00ffa mov r0, #1000 ; 0x3e8
47 806c: eb000059 bl 81d8 <subtract_one>
48 8070: e59f0014 ldr r0, [pc, #20] ; 808c <self_test+0x3c>
49 8074: eb000005 bl 8090 <add_one>
50 8078: e59f000c ldr r0, [pc, #12] ; 808c <self_test+0x3c>
    
```



*Safety through quality*

## PRODUCT BRIEF

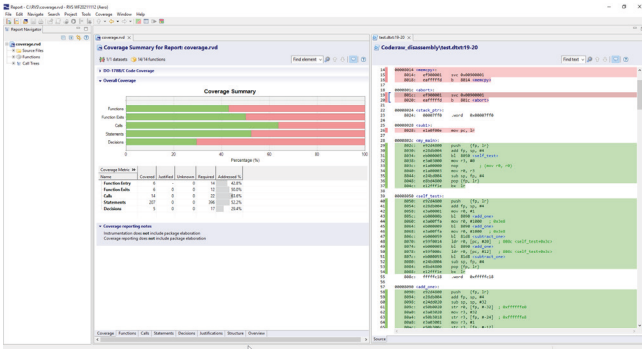
# Zero-footprint coverage analysis with RapiCover<sup>Zero</sup>

# Product brief: RapiCover<sup>Zero</sup>

## RapiCover<sup>Zero</sup>

### How can RapiCover<sup>Zero</sup> help you?

RapiCover<sup>Zero</sup> 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 RapiCover<sup>Zero</sup> 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.

### RapiCover<sup>Zero</sup> use cases

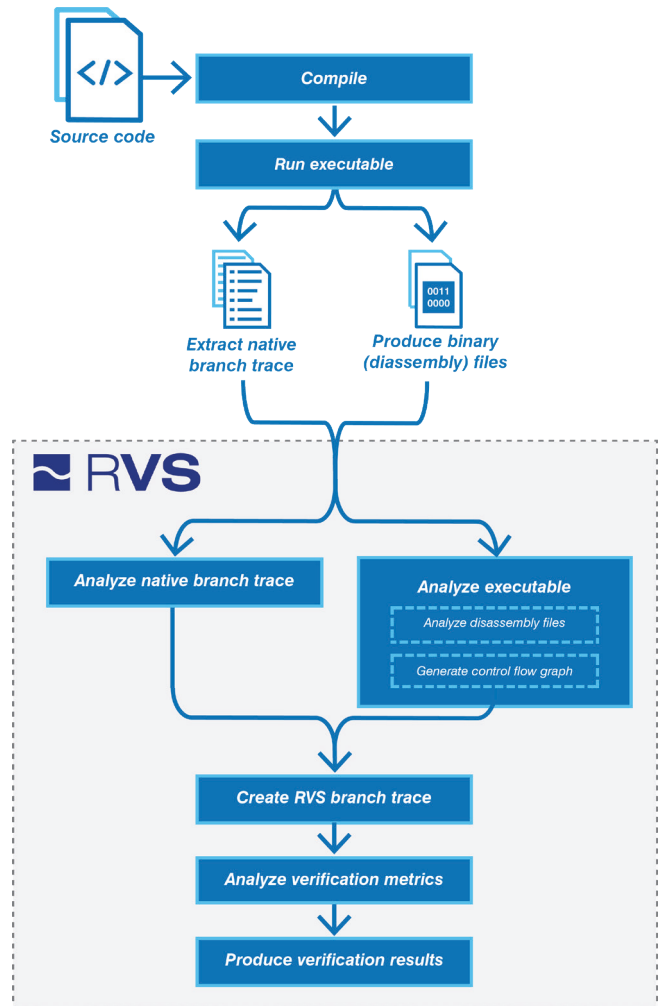
- Structural coverage analysis for third-party libraries.
- Structural coverage analysis with no impact on the code base or development environment.
- View the mapping between source code and object code.
- Address avionics software guidelines for structural coverage analysis: DO-178C, ED-12C, MIL-HDBK-512C, 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 RapiCover<sup>Zero</sup> work?

RapiCover<sup>Zero</sup> 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.**



RapiCover<sup>Zero</sup> verification process

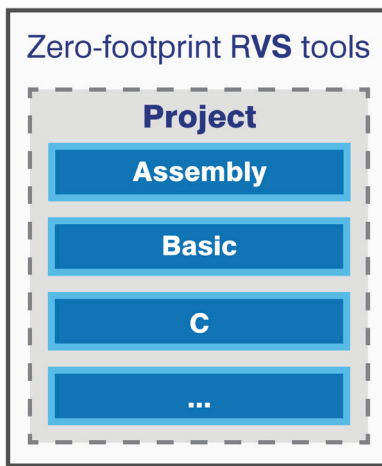
## Key features

### 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



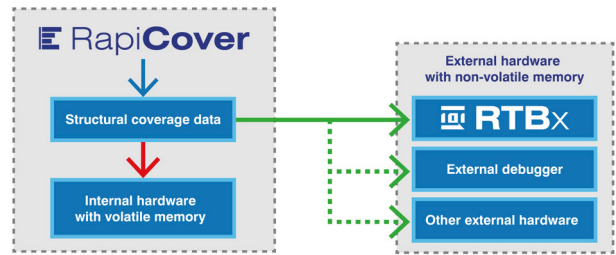
Zero footprint **RVS** tools can be used to analyze software in any language that targets machine code

### 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 RapiCover<sup>Zero</sup> and platform (see *Platform Support Packages*)
- To assess whether a Platform Support Package is available for your platform, see the [compatibility tab on our RapiCover<sup>Zero</sup> product page](#)
- We can develop additional Platform Support Packages to support RapiCover<sup>Zero</sup> analysis for compatible platforms

### Integration support

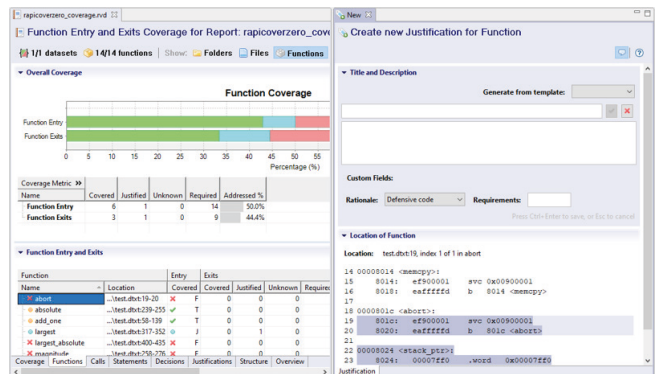
- Automatable testing environment
- Support for very large code bases
- No library/run-time dependencies or dynamic memory requirements
- Coverage analysis across power cycles (subject to hardware requirements)
- Shared integration with other zero-footprint **RVS** tools
- Continuous build servers e.g.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



Analyze structural coverage across power cycles with RapiCover<sup>Zero</sup>.

### 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

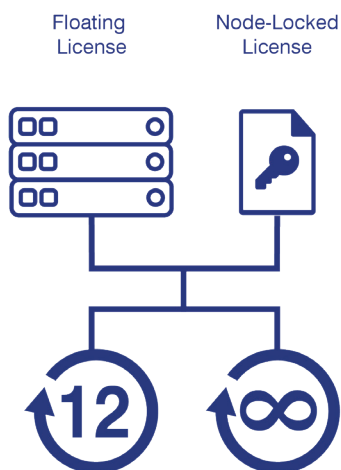
- Summary and detailed results views
- Project and code base insights including code complexity, treemaps, call dependencies, and LOC
- Filter results by subprogram
- Code viewer:
  - View object code alongside source code, where available
  - Color-coded by analysis type and whether code is covered, 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 x86-64 host operating systems:
  - Windows® 10+ and Windows Server® 2016+
  - 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



RapiCover<sup>zero</sup> has flexible licensing options, letting you get the most from the software depending on your needs.

## Should I use RapiCover or RapiCover<sup>zero</sup>?

RapiCover<sup>zero</sup> offers many benefits, but in some cases RapiCover may be more appropriate for you. Consult Table 1 below to decide if RapiCover<sup>zero</sup> or RapiCover is best for you. For more information, contact us at [info@rapitasystems.com](mailto:info@rapitasystems.com).

**Table 1.** Comparison of key RapiCover and RapiCover<sup>zero</sup> features

Feature	RapiCover	RapiCover <sup>zero</sup>
<i>Works without source code</i>	No	Yes
<i>Works without instrumentation</i>	No	Yes
<i>Integration with development environment</i>	Integration needed	No integration needed
<i>MC/DC analysis</i>	Yes	No
<i>Tool qualification support</i>	Yes	Not yet available
<i>Trace size and data processing time</i>	Depends on applied instrumentation	Typically larger trace and longer data processing times
<i>Supported platforms (target, data collection mechanism)</i>	Flexible, almost any platform supported	Requirements on platform (branch trace and context switch information must be available), PSP needed

## Platform Support Packages

To enable RapiCover<sup>Zero</sup> analysis on a specific platform, Platform Support Packages (PSPs) are needed for RapiCover<sup>Zero</sup> 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 RapiCover<sup>Zero</sup> 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 RapiCover<sup>Zero</sup> analysis.

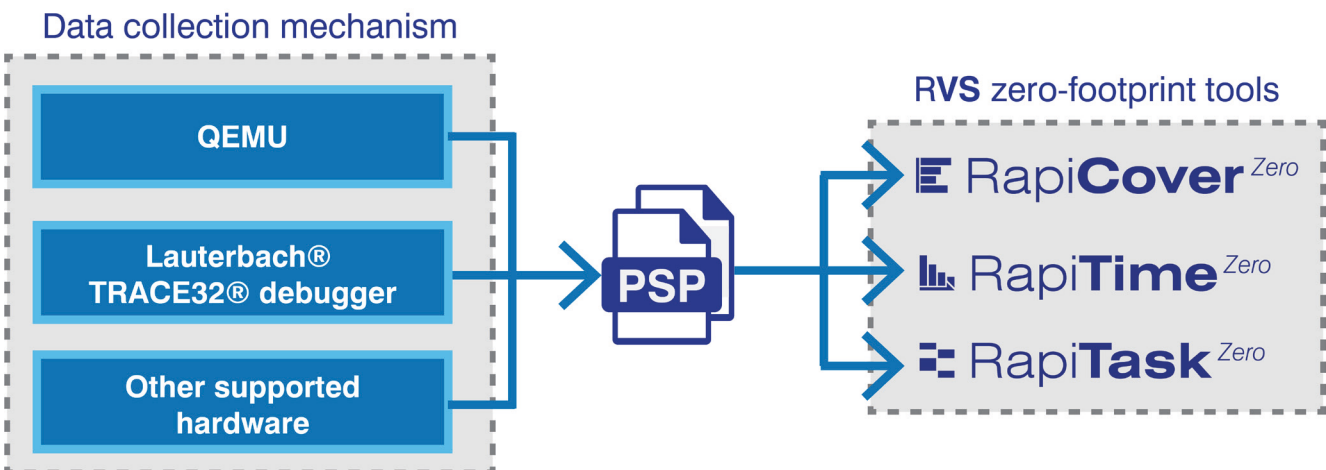
Each PSP is designed to support various components of a platform. These include:

- The compiler(s) used to generate executables to be analyzed by RapiCover<sup>Zero</sup>
- The instruction set of object code to be analyzed by RapiCover<sup>Zero</sup>
- 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 RapiCover<sup>Zero</sup> 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 RapiCover<sup>Zero</sup> when any of the above items are different between two platforms. PSPs that support RapiCover<sup>Zero</sup> analysis also support analysis by RapiTime<sup>Zero</sup> and RapiTask<sup>Zero</sup>. For more information on how Zero-footprint PSPs support analysis by zero-footprint RVS tools including RapiCover<sup>Zero</sup>, 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 RapiCover<sup>Zero</sup> 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](mailto:info@rapitasystems.com).



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



## 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](http://rapitasystems.com/downloads)

## SUPPORTING CUSTOMERS WITH:

### Tools

#### Rapita **Verification Suite:**

Rapi**Test**

Rapi**Cover**

Rapi**Time**

Rapi**Task**

### Engineering Services

#### V&V Services

Integration Services

Qualification

SW/HW Engineering

Compiler Verification

### Multicore verification

#### **MACH**<sup>178</sup>

Multicore Timing Solution

## 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](http://rapitasystems.com)



[linkedin.com/company/rapita-systems](https://www.linkedin.com/company/rapita-systems)



[info@rapitasystems.com](mailto:info@rapitasystems.com)