

Safety through quality

PRODUCT BRIEF

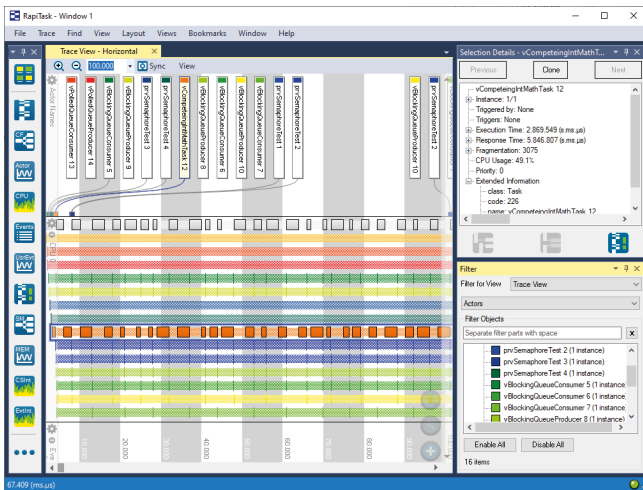
Zero-footprint RTOS event tracing with RapiTask^{Zero}

Product brief: RapiTask^{Zero}

RapiTask^{Zero}

How can RapiTask^{Zero} help you?

RapiTask^{Zero} lets you understand the scheduling behavior of critical software without needing to make any modifications to, or even have access to, your project's source code.



Results displayed in RapiTask^{Zero} user interface

Benefits

Understand the scheduling behavior of critical software without needing:

- Any instrumentation.
- Project source code.
- Any modification to your build system.

RapiTask^{Zero} use cases

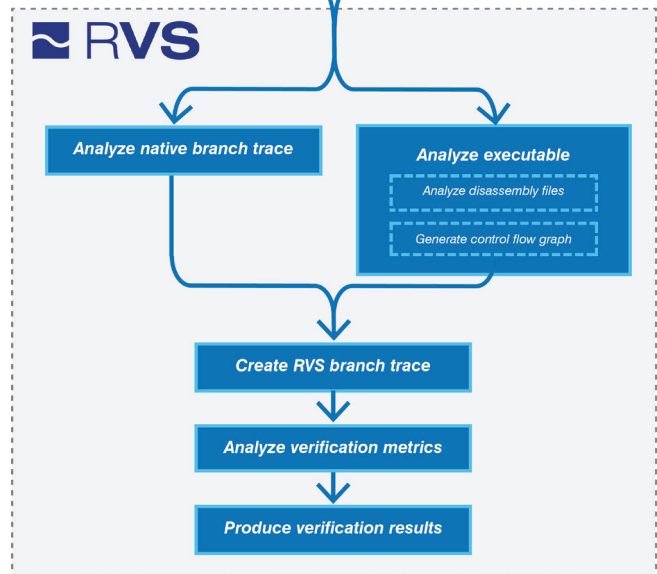
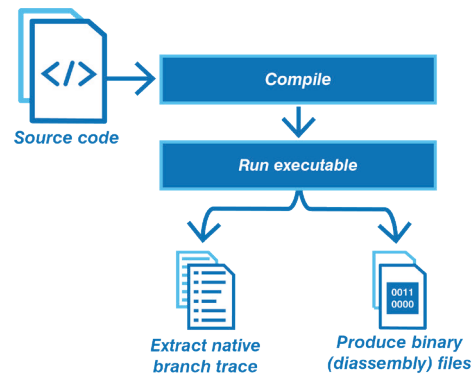
- Understand system scheduling behavior.
- Locate rare timing events.
- Understand system capacity issues.

How does RapiTask^{Zero} work?

RapiTask^{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 scheduling behavior of the executable code 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.



RapiTask^{Zero} verification process

Key features

Task-level timing analysis

- Analyze task-level timing metrics directly from object code
- Calculation of system-level scheduling metrics and related data:
 - Response time
 - Periodicity
 - Jitter
 - CPU utilization (CPU load)
 - Fragmentation
- RTOS-independent scheduling visualization

Language support

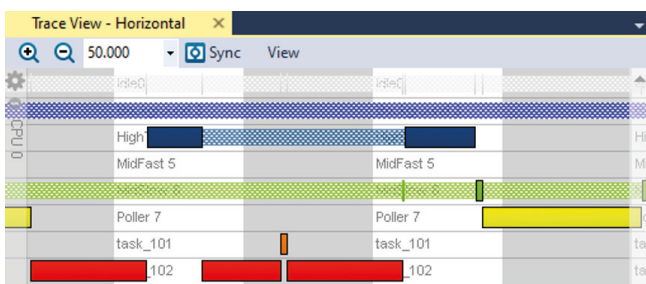
- Any language that targets machine code
- Mixed source languages

Supported platforms

- It must be possible to produce and collect branch trace 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 RapiTaskZero and platform (see *Platform Support Packages*)
- To assess whether a Platform Support Package is available for your platform, see the [compatibility tab on our RapiTaskZero product page](#)
- We can develop additional Platform Support Packages to support RapiTaskZero analysis for compatible platforms

Integration support

- Automatable testing environment
- Support for very large code bases
- No library/run-time dependencies or dynamic memory requirements
- Shared integration with other zero footprint RVS tools
- Multicore support (depending on hardware support)



Identify rare timing events such as priority inversions

Integrated testing environment

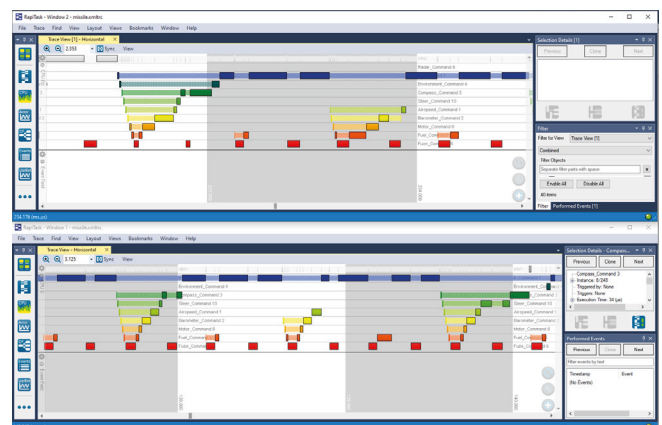
- Invocation timeline charts to help understand timing behavior at a glance
- Custom task coloring
- Hide tasks
- Jump to trace location
- Trace rewind feature to debug timing behavior
- Filter results by subprogram
- Code viewer:
 - View object code alongside source code, where available
- Database-like search function
- Multi-user testing environment

Compatibility

- Runs on host operating systems
 - Windows® 7+ and Windows Server® 2008 R2+
 - 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 transferrable across projects



Compare scheduling behavior of software running on different real-time operating systems

Should I use RapiTask or RapiTask^{Zero}?

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

Table 1. Comparison of key RapiTask and RapiTask^{Zero} features

Feature	RapiTask	RapiTask ^{Zero}
<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>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 RapiTask^{Zero} analysis on a specific platform, Platform Support Packages (PSPs) are needed for RapiTask^{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 RapiTask^{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 RapiTask^{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 RapiTask^{Zero}
- The instruction set of object code to be analyzed by RapiTask^{Zero}
- The native branch trace format generated from the platform – this depends on the mechanism used to generate branch traces, 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 RapiTask^{Zero} are to be run.

Different PSPs are needed to support analysis by RapiTask^{Zero} when any of the above items are different between two platforms. PSPs that support RapiTask^{Zero} analysis also support analysis by RapiCover^{Zero} and RapiTime^{Zero}. For more information on how Zero-footprint PSPs support analysis by zero-footprint RVS tools including RapiTask^{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 RapiTask^{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.



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

Rapita **Verification Suite:**

Rapi**Test**

Rapi**Cover**

Rapi**Time**

Rapi**Task**

Services

V&V Services

Integration Services

Qualification

SW/HW Engineering

Compiler Verification

Multicore verification

CAST-32A Compliance

Multicore Timing Solution

Contact

Rapita Systems Ltd.

Atlas House
York, UK
YO10 3JB

+44 (0)1904 413945

Rapita Systems, Inc.

41131 Vincenti Ct.
Novi, Mi, 48375
USA

+1 248-957-9801



rapitasystems.com



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



info@rapitasystems.com