Rational. software

IBM® Rational® Test RealTime

Highlights

■ *Automates component testing and runtime analysis for host and target from a single testing environment*

■ *Automates creation and deployment of component test harnesses, test stubs, and test drivers*

■ *Profiles memory and performance, analyzes code coverage, and visually illustrates runtime tracing*

■ *Host-based tests easily adapted to different targets without re-writing test procedures*

■ *Test and analyze directly on the target. Supports all common platforms—from an 8-bit microchip to a 64-bit RTOS*

■ *Provides detailed code coverage information required for safety- and mission-critical certification*

IBM Rational Test RealTime is a cross-platform solution for component testing and runtime analysis. Test RealTime was designed specifically for those who write code for embedded, real-time, and other types of commercial software products. Test RealTime allows you to be more proactive in your debugging, enabling you to fix your code before it breaks.

Test, analyze and resolve during development

The best time to find and fix bugs is during development. This is why IBM Rational Test RealTime is focused on developer testing—the kind only you as the author of the code can perform effectively. You need to easily test the components you write and to analyze the reliability and performance of your applications as they run on your host development system.

Rational Test RealTime automates the creation and deployment of component test harnesses, test stubs, and test drivers. With a single click from your development environment, you can also profile memory and performance,

analyze code coverage, and visualize the behavior of your program execution. Fully detailed test and runtime analysis reports are hyperlinked to the relevant source code.

Only IBM Rational Test RealTime combines component testing and runtime analysis into a single, integrated developer-centric testing solution.

Test and debug both host and target

Host-based testing is important for embedded systems development since your target hardware is often not available in a timely manner. But developers cannot measure the quality of their work until they see it execute in the target environment. The situation grows even more complex when you have multiple types of targets that have to be tested.

IBM Rational Test RealTime allows you to test and debug both host and target and to coordinate the two in a meaningful way. When your hardware is not available, you can simulate on the host and test for generic bugs. When the target is there, you can

execute the same tests directly on the operational platform. Test RealTime's versatile Target Deployment Technology allows you to easily adapt your test procedures to any target and build environment without having to rewrite your tests.

Only IBM Rational Test RealTime lets you test and debug on the host and across multiple targets using the same test and analysis procedures.

**Harness the power of model-driven testing**

The Unified Modeling Language (UML) can be used to visually lay out your application's design and test scenarios. IBM Rational Rose®

Technical Developer automates modeling with the UML and provides robust code generation to support Model-Driven Development (MDD) paradigms.

IBM Rational Test RealTime extends MDD to include developer testing activities. We provide a plug-in for Rose Technical Developer that allows developers to invoke runtime analysis features on code generated from UML Models. You can also visualize test case coverage via color-coded state-machines in Rose Technical Developer.

Only IBM Rational Test RealTime integrates the UML across both design and developer testing activities, providing the industry's broadest support for model-driven development.

For more information, please go to http://www.rational.com/products/testrt to learn more, view online demonstrations, and download/order an Embedded Developers Solution CD that includes an evaluation version of Test RealTime.
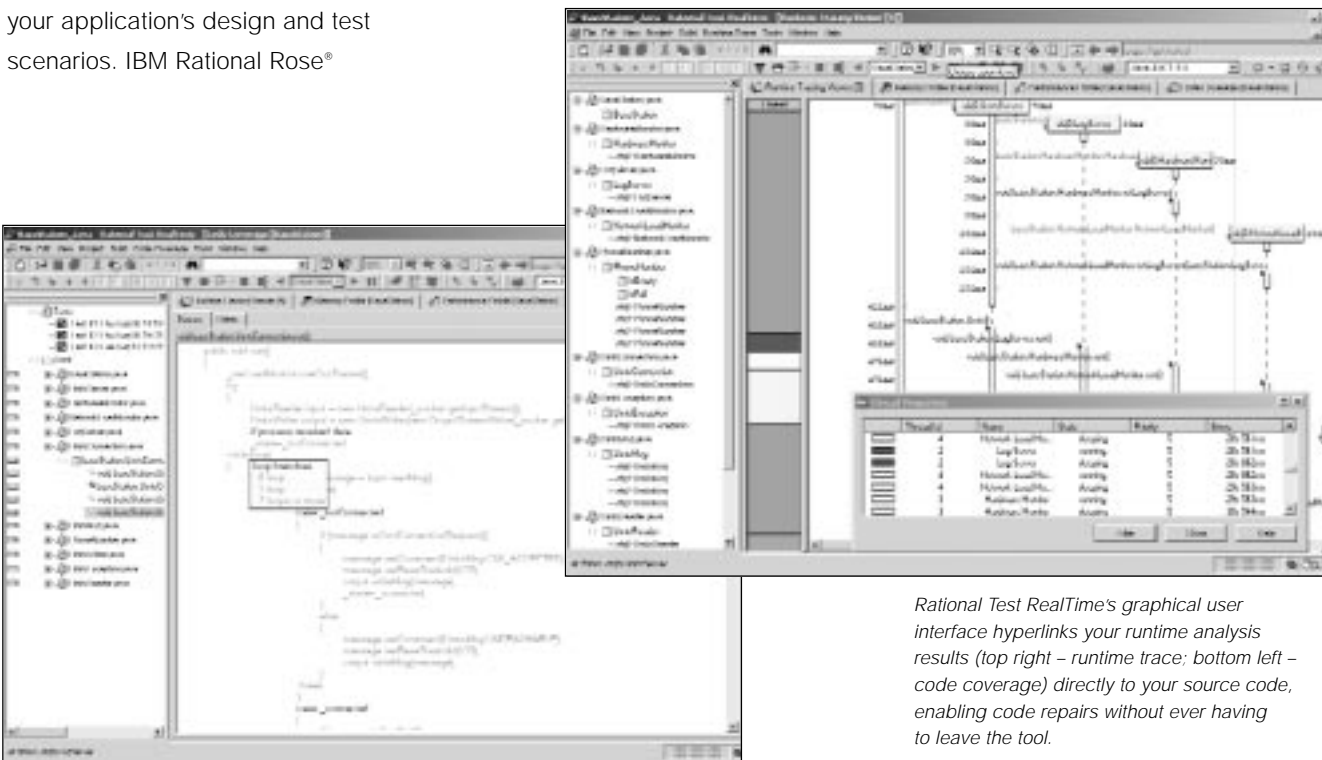
Rational Test RealTime's graphical user interface hyperlinks your runtime analysis results (top right – runtime trace; bottom left – code coverage) directly to your source code, enabling code repairs without ever having to leave the tool.

Features and benefits

Feature	Description	Benefits	Language Support
Component testing	Automates creation and deployment of host and target-based component test harnesses, test stubs, and test drivers	Directly testing on the target increases confidence that actual system will perform as expected	Java, C/C++, Ada
Memory profiling	Illustrates how a program's memory is being consumed and possibly leaked	Identifies the source of eminent or potential memory leaks before they occur, preempting performance issues and program crashes	Java, C/C++
Performance profiling	Identifies application-level performance bottlenecks	Pinpoints areas in the code that can be optimized for their performance	Java, C/C++
Thread profiling	Detects and analyzes multithreading problems, such as deadlocks and race conditions as they occur in programs	Improves the reliability of multi-threaded applications	Java, C/C++
Code coverage analysis	Identifies which portions of a program have not been tested	Avoids delivering code that is executed for the first time by the user or the target system running the application	Java, C/C++, Ada
Runtime tracing	Illustrates thread execution, function calls, and variable values in programs as a function of time via UML sequence diagrams	Developer can go back in time to review how a program behaved after execution has completed	Java, C/C++
System testing	Provides message-based unit and integration testing of C threads, tasks, processes and nodes	Saves time and money by avoiding the construction of dedicated test beds and related non-core-business activities	C
Target Deployment Technology	Provides a versatile, low-overhead technology for enabling target-independent tests and runtime analysis	Your tests won't need to change when your environment does; test script deployment, execution and reporting remain easy to use	Java, C/C++, Ada

IBM®

Supported Platforms

Operating System	Hardware
Windows NT 4.0 SP 6	Intel IA 32
Windows 2000 Pro	Intel IA 32
Windows XP Pro	Intel IA 32
Red Hat Linux 7.2, 7.3 & 8.0	Intel IA 32
SuSE Linux 8.1	Intel IA 32
AIX 4.3, 5.1 & 5.2	IBM RS6000
HP-UX 10.20 and 11	HP PA-RISC
Solaris 2.6, 7, 8 & 9	Sun Sparc

Programming languages
C
C++
Java (J2ME/J2SE)
Ada (component testing & code coverage analysis)

*e*business software

D315E