

Develop Fast, Reliable Code

Rational PurifyPlus

for UNIX

Customers and end-users demand that your code works reliably and offers adequate execution performance. But the reality of your delivery schedule often forces you to sacrifice reliability or performance – sometimes both. And without adequate time for testing, you may not even know these issues exist. So you reluctantly deliver your code before it's ready, knowing good and well that the integrity of your work may be suspect.

Rational Software has a solution for ensuring that your code is both fast and reliable. Rational® PurifyPlus for UNIX combines runtime error detection, application performance profiling, and code coverage analysis into a single, complete package. Together, these functions help developers ensure the highest reliability and performance of their software from the very first release.

Rational® PurifyPlus for UNIX integrates with your existing development processes. It can operate with or without source code. You can choose to incorporate all of its functionality into your daily builds, or use just parts of it on an ad-hoc basis. PurifyPlus for UNIX integrates with the development tools you use for UNIX application development. PurifyPlus for UNIX is easy to adopt, easy to learn, and easy to use.

If you need to develop UNIX code that is both fast and reliable, PurifyPlus for UNIX is your runtime analysis solution.

Rational PurifyPlus quickly pinpoints errors throughout your code.

Automatically Pinpoint Hard-To-Find Bugs

Reliability problems, such as runtime errors and memory leaks in C/C++, can kill a software company's reputation. These errors are hard to find, hard to reproduce, and hardest of all, to explain to customers who discover them. No one intentionally relies on their customer as their "final QA." But without adequate tools, this is the inevitable outcome. Even the best programmers make mistakes in their coding.

Rational PurifyPlus for UNIX automatically finds these reliability errors in C/C++ code that can't be found in any other way. It finds the errors even before visible symptoms occur (such as a system crash or other spurious behavior). And it shows you exactly where the error originated, regardless of how remote from the visible symptom of trouble.

Rational PurifyPlus for UNIX finds these kinds of errors even in third party components where you don't have the source code. This enables you to better manage the quality of code that you deliver but didn't write and don't have easy access to change.

PurifyPlus for UNIX includes the following runtime error checking features:

- Pinpoints runtime and memory-related errors in C/C++ code
- Works on your entire application, including

HIGHLIGHTS

Automatically finds runtime errors in C/C++ code

Quickly isolates application performance bottlenecks

Graphically identifies untested code

Works with or without source code

No code recompiles needed

Instruments entire application, including 3rd party libraries

Defects can be entered into Rational ClearQuest directly from the PurifyPlus GUI



Develop Fast, Reliable Code

third-party components—with or without source

- Quickly analyzes your executables without requiring a recompile
- Finds errors in C/C++ code including heap-related and stack-related errors, pointer errors, memory usage errors and handle errors
- Fully integrated with code coverage analysis features of PurifyPlus for UNIX to identify untested code
- Lets you control the level of error checking per code module

Highlight Performance Bottlenecks

Performance issues—sometimes known as "the final bug"—often get minimal attention, and then only at the very end of the development lifecycle. When the code finally "works," there is a tendency to prematurely deliver to the users—maybe because of schedule pressure, or maybe because performance issues can be hard to find and even harder to resolve.

Rational PurifyPlus for UNIX gives developers a tool that helps developers quickly isolate performance bottlenecks within the application and to understand their impact. Real performance gains come from correcting or improving inefficient algorithms or detailed coding decisions. PurifyPlus for UNIX offers repeatable performance data (not just based on sampled data) and clear, concise views of performance profile data to help developers identify the algorithms that are eating away at performance. The PurifyPlus Call Graph, in particular, highlights algorithmic performance data per thread basis.

The Rational PurifyPlus graphical display of performance data makes it easy see which functions to 'tune' first, because thicker lines indicate more time consumed.

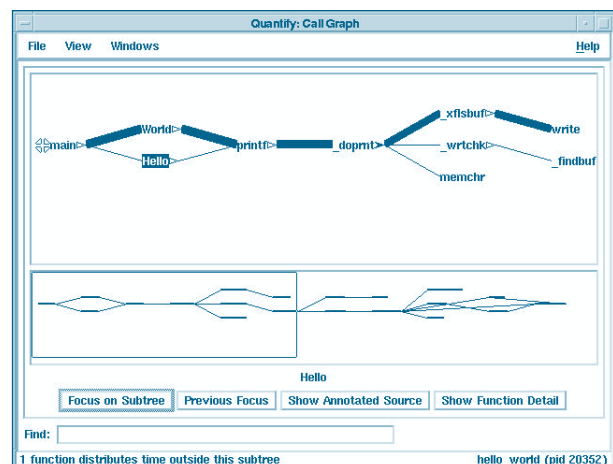
Performance issues are not necessarily related to algorithmic errors. An algorithm may be functionally correct and accurate, but simply inefficient in its implementation. There often exists multiple ways to code an algorithm so that it is functionally equivalent to the original yet notably faster in computer execution. PurifyPlus for UNIX identifies performance bottlenecks, allowing the designer or programmer to determine if such optimizations are in order.

When working on performance issues, it is critical to be able to measure improvements analytically. The PurifyPlus "Diff" function enables developers or performance engineers to validate the impact of changes on improvements in performance and scalability, even on small test cases.

Rational PurifyPlus for UNIX provides a global, graphical view of the performance data you need to quickly identify performance problems from the beginning of the development process. PurifyPlus for UNIX offers truly repeatable and accurate results because it uses actual clock or machine cycle counts rather than the sampling methods used by traditional performance profiling tools. This reduces the difficulty and guesswork of performance tuning because it helps you to analyze precise performance data and immediately focus in on the changes that will have the greatest performance impact.

PurifyPlus for UNIX includes the following application performance analysis features:

- Highlights application performance bottlenecks in C/C++ code and Solaris-based Java code



- Quickly analyzes your executables without requiring a recompile
- Provides accurate, thorough and repeatable performance data on all functions, with or without source code
- Allows the comparison of multiple versions to immediately highlight the precise impact of changes on application performance
- Lets you control the level of performance data collected per module

Avoid Shipping Untested Code

Any testing tool is only as good as the coverage provided by its test suite. But judging the quality of a test suite can be difficult. In particular, it is hard to know if you have tested all paths in which the code can execute, and if not, what sections you have missed. You don't want your customer to be the first to exercise any portion of your application. But without exhaustive testing efforts, that often happens.

Rational PurifyPlus for UNIX solves this problem by allowing developers to see exactly what parts of their code have been tested, and more importantly, what parts have not been tested. A color-coded display shows the developer code coverage analysis results on a function or line-by-line basis, even showing which code fragments on a particular line were (or were not) tested.

Verifying test suite coverage (and improving on it where needed) is the best possible way to maximize the benefit of using Rational PurifyPlus for UNIX. PurifyPlus for UNIX makes it easy for you to test your code coverage, whether you are checking for run-time errors, investigating performance issues, or both. Verifying test suite coverage is the final step in guaranteeing that your customer will receive the highest quality code the first time, every time.

PurifyPlus for UNIX includes the following code coverage analysis features:

- Identifies untested code throughout C/C++ applications and Solaris-based Java applications
- Quickly analyzes your executables without requiring a recompile
- Analyzes your entire application including components, with or without source code
- Presents coverage statistics for each run of the executable, in an intuitive color-coded display
- DIFF and MERGE capabilities automatically allow you to compare and merge coverage data from multiple runs of the same executable for an aggregate view of coverage data
- Fully integrated with the memory error and memory leak detection features of PurifyPlus for UNIX
- Lets you control the level of code coverage data collected per module

Instantly Become More Productive

With Rational PurifyPlus for UNIX, you benefit not just from the power of the best-in-class features described above, but also from the powerful integrations these tools offer. Everything you need is integrated into one product. PurifyPlus for UNIX is designed to work within your existing development environment, allowing you to develop fast, reliable code more productively.

Rational PurifyPlus for UNIX integrates with Rational ClearQuest® for UNIX for reporting and managing defects. You can submit defects directly from the PurifyPlus for UNIX GUI, and then track them in ClearQuest.

Incrementally Improve Code Reliability or Performance

Rational PurifyPlus for UNIX consists of Rational Purify® for UNIX, Quantify® for UNIX, and PureCoverage® for UNIX, packaged together at an attractive price with a common install and common licensing for convenience. Rational also sells each of these individual tools separately for organizations that want to begin improving their code reliability or performance more incrementally.

Runtime Analysis for Embedded Systems

PurifyPlus is designed to be used on a host computer. Embedded application developers concerned with memory use or performance bottlenecks can use PurifyPlus during simulation execution on the host. For target system runtime analysis, Rational offers memory leak and performance bottleneck analysis in a separate product called Rational Test RealTime. See <http://www.rational.com/products/testrt/index.jsp> for details.

Unify Your Team With Rational Suite

Rational PurifyPlus for UNIX is included in Rational Suite® DevelopmentStudio™ for UNIX. The Rational Suite product family provides a comprehensive development platform that will unify your team, optimize individual productivity, and simplify adoption of the full Rational solution.

Accelerate Success with Rational Services

Like all Rational tools, Rational PurifyPlus for UNIX is supported by an extensive, worldwide service organization. Explore more than a thousand articles, white papers, courses, and artifacts online at the Rational Developer NetworkSM. Build your team's capability through expert consulting and technical support services tailored for Development Teams using Rational tools. And take advantage of more than 60 courses available from Rational University to speed technology deployment and accelerate your project's delivery. Available where and when needed, Rational Services improve self-sufficiency as they build a foundation for continuous software development improvement.

The screenshot shows the PureCoverage application window with a menu bar (File, View, Actions, Adjustments, Help) and a toolbar. Below the toolbar is a table with the following data:

| Sorting order: Adjusted unused lines | Runs Calls | FUNCTIONS | | | ADJUSTED LINES | | | ADJS total |
|---|------------|-----------|------|-------|----------------|------|-------|------------|
| | | unused | used | used% | unused | used | used% | |
| ▼ Total Coverage | | 0 | 3 | 100% | 0 | 9 | 100% | 0 |
| ▼ /usr/home/pat/example/ | | 0 | 3 | 100% | 0 | 9 | 100% | 0 |
| ▼ hello_world.c | 2 | 0 | 3 | 100% | 0 | 9 | 100% | 0 |
| display_hello_world | 1 | | used | | 0 | 2 | 100% | 0 |
| display_message | 1 | | used | | 0 | 2 | 100% | 0 |
| main | 2 | | used | | 0 | 5 | 100% | 0 |

The code coverage analysis features of Rational PurifyPlus highlight which portions of your code have not been tested.

SPECIFICATIONS

Supported Environments

Sun Platform

- Sun Solaris 2.6, 7, 8, 9
- Sun SPARC systems
- Languages: C, C++ (all features); Java (performance profiling & code coverage analysis only)
- Full 64 bit "wide-mode" support for memory error & leak detection and performance profiling on Solaris 7 and above; 32 bit support for code coverage analysis
- Compiler support through Sun ONE Studio 7 (Forte CC 5.4 compiler), and GCC through v3.1 (32-bit only)

HP Platform

- HP-UX 10.20, 11.0, 11i (11.11)
- HP 9000 Series 700 and Series 800 workstations
- Languages: C, C++
- Full 64 bit "wide-mode" support on HP-UX 11.0 and 11i (11.11)
- Compiler support for HP aCC 3.31, 3.33 and 3.34; GNUPro 98r2; and GCC through v3.1
- Support for ld/dld version 11.30

Rational Software

Dual Headquarters

18880 Homestead Road
Cupertino, CA 95014

20 Maguire Road
Lexington, MA 02421

Toll-free: (800) 728-1212
e-mail: info@rational.com
Web: www.rational.com

International Locations:
www.rational.com/worldwide