PARASOFT®

PARASOFT Jtest®

Parasoft® Jtest® is an integrated development testing solution for applications written in Java. Jtest automates a broad range of practices—including static code analysis, unit testing, coverage analysis, code metrics, and more—that improve development team productivity and software quality. Jtest can also collect and merge coverage data across testing techniques, such as manual and automated functional testing, by leveraging tools such as Parasoft SOAtest—enabling end-to-end functional and load testing for today's complex distributed applications and transactions.

Tests can be run directly from the IDE or as part of an automated process. To promote rapid remediation, each problem detected is prioritized based on configurable severity assignments, automatically assigned to the developer who wrote the related code, and distributed to his or her IDE with direct links to the problematic code and a description of how to fix it.

Automate Code Analysis for Compliance

Benefits

Parasoft's customers, including the majority of the Fortune 500, rely on Jtest for:

- Preventing defects that impact application security, reliability, and performance
- Complying with internal or regulatory quality initiatives
- Ensuring consistency across large and distributed teams
- Increasing productivity by automating tedious yet critical defect-prevention practices
- Successfully implementing popular development methods, such as TDD and Agile

Increase programming proficiency across the organizations and prevent entire classes of errors that have bugged the software industry.

Static code analysis	Facilitates regulatory compliance (FDA, PCI, etc.). Ensures that the code meets uniform expectations around security, reliability, performance, and maintainability. Eliminates entire classes of programming errors by establishing preventive coding conventions.
Data flow static analysis	Detects complex runtime errors related to resource leaks, exceptions, SQL injections, and other security vulnerabilities without requiring test cases or application execution.
Metrics analysis	Identifies complex code, which is historically more error-prone and difficult to maintain.
Unit test execution	Enables the team to verify functional and non-functional requirements to reduce the duration and cost of downstream debugging.
Coverage analysis	Assesses test suite efficacy and completeness using a multi-metric test coverage analyzer. This helps demonstrate compliance with test and validation requirements.
Team deployment and workflow	Establishes a sustainable process that ensures software verification tasks are ingrained into the team's existing workflow and automated so team members can focus on tasks that truly require human intelligence.
Error assignment and distribution	Facilitates error review and correction. Each issue detected is prioritized, assigned to the developer who wrote the related code, and distributed to his or her IDE with direct links to the problematic code.

These core capabilities are also available for C, C++, .NET languages.

Centralized reporting	Ensures real-time visibility into quality status and processes. This helps managers assess and document trends, as well as determine if additional actions are needed for regulatory compliance.
Continuous "On-the-fly" static analysis	Automatically run static analysis in the background as developers review, add, and modify code. This helps the team identify and fix problems as soon as they are introduced.

Key Features

- Built-in support for Google Android, Spring, Hibernate, Eclipse plug-ins, TDD, JDBC, EJBs, and more (mobile, embedded, Java EE...)
- Integrates with Parasoft SOAtest for end-to-end functional and load testing for web, SOA, and cloud development.
- Exposes runtime defects that occur as the application is exercised by unit, manual, or scripted tests–including race conditions, exceptions, resource leaks, and security attack vulnerabilities
- Without requiring execution, identifies execution paths that can trigger runtime defects
- Checks compliance to configurable sets of over 1000 built-in static analysis rules for Java
- Provides templates for OWASP Top 10, CWE-SANS Top 25, PCI DSS, and other security static standards
- Allows easy GUI-based customization of built-in rules
- Identifies and prevents concurrency defects such as deadlocks, race conditions, missed notification, infinite loops, data corruption other threading problems
- Automatically creates robust low-noise regression test suites–even for large code bases
- Execute tests created in any open-format unit testing tool, including out-of-the-box support for JUnit- and TestNG– based testsIntegrates and extends manually-written unit test cases
- Continuously executes the test suite to identify regressions and unexpected side effects
- Performs runtime error detection as tests execute
- Parameterizes test cases for use with varied, controlled test input values (runtime-generated, user-defined, or from data sources)
- Monitors test coverage with multiple metrics
- Tracks code coverage from manual tests and test scripts
- Steps through tests with the debugger
- Tests individual methods, classes, or large, complex applications
- Calculates metrics such as Inheritance Depth, Lack Of Cohesion, Cyclomatic Complexity, Nested Blocks Depth, Number Of Children
- Identifies duplicate and unused code
- Shares test settings and files team-wide or organization-wide
- Generates HTML, PDF, XML, and custom reports
- Tracks how test results and code quality change over time
- Provides GUI (interactive) and command-line (batch) mode

Supported Environments

Operating System

Windows: 7, 8 or 10 (x86 or x86_64), Server 2008, or Server 2012.

- Linux: 2.6 or newer kernal with glibc 2.3 or newer (x86 or x86_64)
- Solaris: Solaris 10 (SPARC)
- Mac: OS X 10.9 or higher

Hardware

- Intel® i5 2.0 GHz or higher recommended
- 8 GB RAM minimum; 16 GB RAM recommended

Software

- Eclipse 3.7, 4.2, 4.3, or 4.5.
- IBM Rational Application Developer 7.0-8.0
- JRE 1.5 or higher
- Integration with Ant, Maven, Gradle, Jenkins, CruiseControl, other build and release tools
- Integration with most popular source control systems
- Open Source Control API, which allows teams to integrate with other source control systemws

PARASOFT

USA PARASOFT HEADQUARTERS / 101 E. Huntington Drive, Monrovia, CA 91016
Phone: (888) 305-0041 / Email: info@parasoft.com