

## IBM Rational Rhapsody testing solutions

### Highlights

- *Helps improve quality and productivity by using an automated testing solution.*
- *Brings the benefits of modeling to the testing process, extending MDD to include model-driven testing for production code.*
- *Assists organizations striving to close the productivity gap between development and testing.*
- *Aids model coverage by generating relevant combinations of inputs to achieve high coverage of the code inside the model, complementing your requirements-based generated test cases.*
- *Offers a complete approach that can help drive down costs while improving quality.*



By harnessing the power of IBM® Rational® Rhapsody® model-driven development (MDD), some software developers and systems engineers have improved their productivity—yet code-centric methods still dominate the testing process. Testing has a reputation for consuming the majority of development time, but new IBM Rational solutions are helping to close the gap between development and

testing. With the latest IBM Rational Rhapsody TestConductor Add On and IBM Rational Rhapsody Automatic Test Generation Add On (ATG) products, IBM Rational software helps close the productivity gap between development and testing by bringing the benefits of modeling to the testing process, thus extending MDD into the testing domain to include model-driven testing for production code.

## Defining tests

To facilitate unit and integration testing, the Rational Rhapsody TestConductor Add On product is designed to automatically generate a test architecture, which is a graphical Unified Modeling Language (UML) model of a code-based test harness for portions of the design that need to be tested. Test cases are defined in the context of a test architecture, and based on the design requirements testing need, a test case can be defined as code, as a flowchart or as a scenario.

While code-based test cases help users to code the “pure” behavior of the test case, a flowchart helps them to do the same thing graphically. In both cases, the test architecture integration is provided automatically. While this does not change the way most of us are used to describing test cases, it offers major improvements because it helps to leverage information captured in the model. Scenarios, the third type of test cases, are often used to graphically capture system-level requirements as well as derived requirements, and these are also converted into test cases with a click of a button.

Another way to find test cases is to use the Rational Rhapsody Automatic Test Generation Add On environment, which iterates the model via model

element coverage, covering states, transitions, operations, and event generation. Using ATG, you can get model coverage that generates relevant combinations of inputs in order to achieve high coverage of the code inside the model, helping to complement your requirements based generated test cases.

Engineers know that it is important to get code-based results when running your model-based test case. The Rational Rhapsody TestConductor Add On integration with the IBM Rational Test RealTime helps you run tests and automatically get complete, textual and graphical code-based tests results while running on the host or target.

## Test execution

The Rational Rhapsody TestConductor Add On solution can execute a combination of test cases by launching the generated Rational Rhapsody application on the host or the target platform. Progress information is displayed in run time. To facilitate design debugging (and test case debugging), you can step through test case execution and perform scenario-based testing, even generating a new sequence diagram to reflect a concrete execution of a selected instance of a sequence diagram, helping to simplify the debugging process.

## Analyzing test results

Failure of a code- or flowchart-based test case can be traced to the offending error. Additionally, assertions can be inserted to validate and trace the execution of the test case. When a scenario-based test case fails, users can generate a multicolored sequence diagram to display the set of messages generated and successfully monitored, including the “offending” message and the expected sequence. Not only does this help save time in identifying and understanding the problem, it also helps make correcting the design very easy. You simply re-execute the test in conjunction with Rational Rhapsody animated statecharts — watching the design as it executes, step by step, until it reaches the problem area. With the integration to Rational Test RealTime these same scenarios can be reused to help get code coverage, memory profiling, and performance analysis results easily organized inside the model. This complete approach can help you drive down costs while improving quality, leveraging the design by continuously testing it against requirements as it is developed.

## How it works

The Rational Rhapsody testing solution consists of the Rational Rhapsody TestConductor Add On and Rational Rhapsody Automatic Test Generation

Add On products. The Rational Rhapsody TestConductor Add On solution is a UML-compliant testing environment for realtime embedded applications. By analyzing your design, the Rational Rhapsody TestConductor environment can help build the test context automatically, including the test architecture that is required for each class (basic or composite) that needs to be tested. Test cases that are described as code are compiled and built along with the code that the Rational Rhapsody tool generates from the test architecture. Similarly, test cases that are described as flowcharts are converted “on the fly to code” and built with the code that reflects the test architecture. Test cases that are described as sequence diagrams are converted into code where the “inputs” to the system under test (SUT) are

driven from the test case and so are the resulting messages that need to be observed.

The Rational Rhapsody Automatic Test Generation Add On solution offers a superior capability: by analyzing the design, it generates sequence diagrams that drive the design through many paths with a goal of helping to maximize the coverage of the design (every state covered at least once, every transition fired at least once, etc.). However, since the automatically generated test cases look and behave exactly as the sequence diagrams that a human tester would write, the Rational Rhapsody ATG-generated test cases can be executed using Rational Rhapsody TestConductor Add On software, along with any combination of test cases, virtually regardless of their origin.

The ability to capture test cases in multiple formats facilitates a gradual process migration from code-based to model-driven testing, helping to improve return on investment as your testing process improves. Furthermore, since model-driven testing leverages the design and helps enable developers to continuously test their design against requirements throughout the development process, the solution can aid in reducing development time and costs while helping improve product quality.

### For more information

To learn more about IBM Rational Rhapsody software, contact your IBM representative or IBM Business Partner, or visit:

[ibm.com/software/rational](http://ibm.com/software/rational)

Feature	Functional benefit
Automatic test architecture generation	Helps improve testing process by leveraging design
Define your test case behavior using plain code, flowcharts or sequence diagrams	Helps increase productivity by automating mundane testing tasks
Use sequence diagrams requirements for requirements-based testing	Enhances quality by testing earlier in the development cycle, mitigating problems when they are less costly to fix
Generate and use recorded sequence diagrams for regression testing	Boosts communication among stakeholders and the development team by capturing test cases graphically as part of the development model
Automatically generate test cases to support higher coverage	Assists in automating and improving the coverage of your testing processes
Execute test cases to generate pass/fail reports	Aids in documenting and streamlining your test case reporting procedures
Support for code coverage, memory profiling, and performance analysis	Helps enable your organization to leverage best-in-class testing processes



© Copyright IBM Corporation 2009

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
June 2009  
All Rights Reserved

IBM, the IBM logo, ibm.com, Rational and Rhapsody are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this document is provided for informational purposes only and provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. Without limiting the foregoing, all statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.