

Automated Testing

As experienced mainframe personnel retire, and teams face shortened lead times, next-generation developers must be able to quickly and confidently make changes to mainframe programs. **Topaz® for Total Test**, Compuware's automated testing tool, enables developers and testers at all skill levels to test programs and subprograms almost immediately after updating a piece of code to ensure unintended changes weren't introduced into the codebase.

VIRTUALIZED AND NON-VIRTUALIZED TESTING

Topaz for Total Test can test a mainframe program in two ways:

- **Virtualized:** Topaz for Total Test can virtualize testing, meaning all subprogram calls, data accessed and API calls are virtualized or stubbed out and the program is tested in complete isolation from external dependencies. This type of testing is typically done by a developer.
- **Non-virtualized:** A program may also be tested in a non-virtualized manner, where testing is executed in a real environment with real subprograms and data. This ensures the program not only works as expected, but also that it works correctly with all required subsystems available. This type of testing is typically handled by developers and QA staff.

UNIT, FUNCTIONAL, AND INTEGRATION TESTING IN ONE SOLUTION

Topaz for Total Test enables automated virtualized and non-virtualized testing for all testing types, ranging from unit and functional to system, integration, and regression testing.

AUTOMATED TESTING: EXECUTION AND CREATION

Topaz for Total Test automates the execution of virtualized and non-virtualized tests within CI/CD pipelines and is directly supported by pipeline technologies such as **Jenkins**, **Digital.ai Release**, and **CloudBees Flow**. Other pipeline technologies are supported as well through integration with the Topaz Command Line Interface (CLI). By creating and executing virtualized and non-virtualized tests of a program or a group of programs, developers will ensure they have a regression test suite that covers all foreseeable failure scenarios.

Topaz for Total Test also automates the creation of virtualized and non-virtualized tests saving developers valuable time that would normally be spent on manually creating tests and test assets.

CODE COVERAGE

For both virtualized and non-virtualized testing, integration with Compuware **Xpediter**® Code Coverage allows developers to collect and analyze code coverage metrics after test case execution to understand what part of the program code was covered by a specific test case. Actively using code coverage allows developers to build a comprehensive suite of tests that cover important program logic.

VIRTUALIZED TESTING

A developer can automatically create a virtualized test case for a program in just a few steps. By simply clicking a Record button, the developer can submit a job that will execute the program to be tested and automatically record a test case and related virtualized test assets, including program stubs and test data from Db2, IMS, CICS, and datasets. Test assets are saved so they can be used repeatedly in later testing to virtualize real data. Topaz for Total Test virtualizes (stubs out) program-to-data, program-to-subprogram calls, as well as IMS and CICS API calls and returns data for only what is programmed into the test.

By verifying that programs consistently return the same data before and after code changes, Topaz for Total Test facilitates regression testing, enabling developers to determine if any bugs were inadvertently introduced when code changes were made.

NON-VIRTUALIZED TESTING

Topaz for Total Test supports non-virtualized testing of load modules on an LPAR in batch and CICS using real data and subsystems. Programs can be written in COBOL, PL/I, or Assembler. Non-virtualized tests are defined in a test scenario, and the user can either record a test or manually specify input and expected output data for the linkage section of the module.

Developers use a test scenario to execute non-virtualized program testing and for testing complete batch jobs. Test scenario elements can be used to create a sequence of steps (e.g. set up data in datasets and Db2) before testing programs; execute programs by providing input data and verifying output data; and finally verify data created by the program in Db2 and datasets.

Test Data Management

Users can automate a test data setup to ensure data is consistent and accurate before running a test case for a non-virtualized test scenario. Through an integration with **Topaz for Enterprise Data**, users may define a step in the test scenario that will execute a load specification containing extracted data for Db2 tables. The execution will update Db2 tables with the required data and ensure that subsequent steps in the test scenario use the correct data.

FACT SHEET**Verifying Datasets**

An important part of testing a program is ensuring the output data is as expected. Developers have two options to verify datasets in a non-virtualized test scenario. They can use a built-in dataset verification element that compares two datasets record by record, as well as, verify content of records by writing simple expressions for validating specific fields in a record. Alternatively, they can save time by adding a **File-AID**® Compare step to the test scenario to compare the datasets.

A test scenario, which can be used to test complex batch jobs, can also be used to create an integration test of multiple program interactions. For example, it can be set up to execute a sequence of different programs; use output data from one program as input data to the next; and verify response data and data produced in Db2 and datasets between each step.

COHESIVE USER EXPERIENCE

Topaz for Total Test enables developers to run virtualized and non-virtualized tests from the same test scenario—using one editor and one execution dialog. With this combined test scenario, JCL is now automatically generated to run the tests. Test scenarios can be combined into test suites that are automatically executed in CI/CD pipelines with Jenkins or other pipeline technologies. Test results can be captured along with other details in one report and housed in a shared repository.

TEST ASSERTIONS

Default test result assertions enable comparisons between the expected results the developer or tester defines in the test and the actual results returned by the program.

After the test executes, test assertions are evaluated to determine if they pass or fail. Test assertions not only provide test results, but also include the reason for the failure, greatly speeding test failure resolution.

TEST EXECUTION FLEXIBILITY

Developers can execute tests on different LPARs based upon available test environments. If a developer is using test assets and wants to execute the test on another LPAR, the test assets are moved to the other LPAR as part of the test case for execution.

Alternatively, you can choose from a simple list of LPARs with test runners. Simply pick the LPAR and run the test.

TEST DATA VIRTUALIZATION

Topaz for Total Test simplifies and accelerates testing of large COBOL programs by automatically virtualizing data for Db2 databases, VSAM, and QSAM data files, as well as subprogram calls. It also supports IMS BMP/MPP and batch DL/I command virtualization, eliminating the need to have an IMS system when testing an IMS program. Similarly, CICS commands can be virtualized so CICS tests are isolated from actual CICS regions.

DEVOPS TOOLCHAIN INTEGRATIONS

You can easily integrate test assets into a DevOps toolchain to enable virtualized and non-virtualized testing within the Jenkins automated build-test-deploy process. A Jenkins plugin facilitates seamless configuration of test execution in a Jenkins workflow for enterprise DevOps efficiency.

An integration with Digital.ai Release enables users to integrate a Topaz for Total Test virtualized test scenario and/or test suite into a DevOps pipeline. Similarly, users can run virtualized test scenarios and test suites within a DevOps pipeline orchestrated by CloudBees Flow.

Test and code coverage results can be automatically fed into [SonarSource SonarQube](#) for analysis, enabling IT teams to understand the scope and effectiveness of their testing as code is promoted towards production. (Code coverage metrics provide insight into which lines of code have or have not been executed, and what percentage of an application has or has not been tested.) Using SonarQube, developers can also see test results along with code quality trends across multiple platforms throughout the DevOps process.

A Jenkins plugin automatically generates Topaz for Total Test pass/fail results into a format SonarQube can display, enabling DevOps teams to see how testing progresses after every build. SonarQube Quality Gates on test results can be used to determine whether to proceed with a workflow or stop and fix the code before continuing.

Intelligent Automated Test Execution

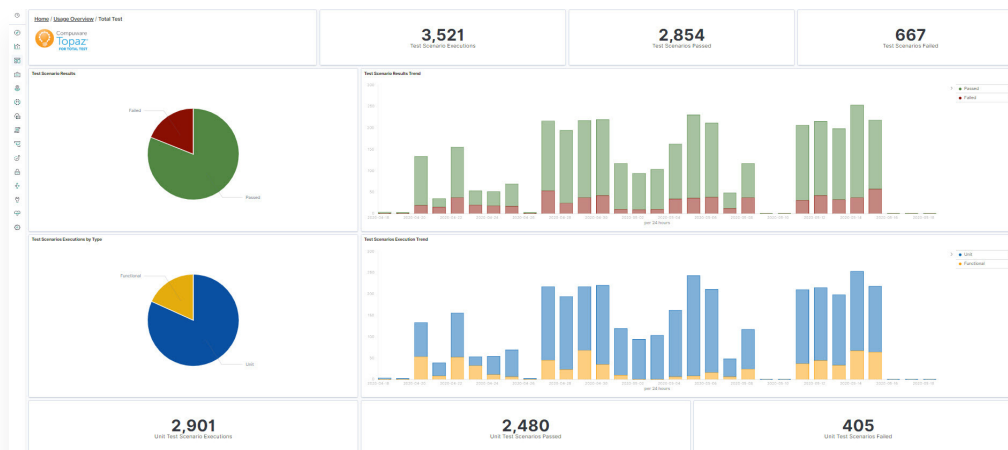
When setting up an automated CI/CD pipeline to automate the build, deployment, and testing of mainframe applications, Topaz for Total Test can intelligently execute only the test cases related to changed programs. By selectively choosing which tests to run, the pipeline runs faster, speeding software delivery and saving on mainframe resources.

This approach works especially well when [ISPW](#)[®] is used as a source control and build mechanism for mainframe source code. When a developer promotes a change in ISPW, it triggers a pipeline to run, and the pipeline will automatically know what programs were changed. Topaz for Total Test then uses that information to select and execute related test cases.

FACT SHEET

ANALYTICS

Mainframe testing activities can be captured in a detailed Compuware **zAdviser** dashboard enabling Topaz for Total Test users to assess the efficacy of their testing efforts. Likewise, development managers can quickly measure adoption of Topaz for Total Test, what is being tested, success and failure rates, as well as whether tests are being leveraged within a DevOps pipeline—intelligence critical to helping teams continuously improve mainframe software development and delivery outcomes.



zAdviser dashboard showing testing activities at a macro level.

FACT SHEET**THE VALUE OF TOPAZ FOR TOTAL TEST**

- Enables developers and testers to automate virtualized and non-virtualized testing, covering unit, functional and integration testing with one easy-to-use tool.
- Tests with virtualized (no external dependencies) and/or non-virtualized (real environment) subprogram calls and data.
- Intelligently executes only test cases related to changed programs in a CI/CD pipeline.
- Ensures test data consistency and accuracy through an integration with Topaz for Enterprise Data.
- Reduces dependency on specialized knowledge of mainframe veterans by empowering novice developers to work with and validate mainframe code changes quickly and accurately.
- Ensures quality of business-critical mainframe code by giving organizations confidence to make large mainframe code changes.
- Empowers teams to put code changes into production more quickly to better respond to changing business needs.
- Ensures continuous code quality management across platforms through an integration with SonarQube.

Learn more at compuware.com/topaz-for-total-test.

Learn more about the Topaz Suite at compuware.com/topaz.

The Mainframe Software Partner for the Next 50 Years

Compuware, a BMC company, empowers the world's largest companies in their ongoing transformation to an Autonomous Digital Enterprise by delivering innovative software that enables IT professionals with mainstream skills to develop, deliver, and manage mainframe applications with ease and agility.

[Learn more at compuware.com](https://compuware.com).