

## BMC AMI DevX TOTAL TEST

# BMC AMI DevX Total Test 20.x: Fundamentals Automated Testing

[Learning Path >](#)

Course Code: CPKT-TSTG-F2012

**Modality**

Instructor-Led Training (ILT)

**Duration**

4 Days

**Applicable Versions**

- BMC Compuware Topaz for Total Test version 20.10, 20.11, 20.12, 20.13
- BMC AMI DevX Total Test 20.14

**Target Audience**

- Developers (Testers)

## Course Overview

BMC AMI DevX Total Test (formerly known as BMC Compuware Topaz for Total Test) is an automated testing solution that enables developers and testers to test mainframe programs and subprograms almost immediately after updating a piece of code. Developers need to test programs frequently and repeatedly to verify their changes while remaining independent from the environment. BMC AMI DevX Total Test enables automated virtualized and non-virtualized testing by allowing virtualization of the environments and external data sources.

This course teaches you how to create test projects in Total Test, execute tests, extend test scenarios, rebaseline tests, and prepare and validate tests. The course trains you on how to perform automated virtualized and non-virtualized tests through classroom training sessions and hands-on lab experience.

This course was created with Topaz for Total Test version 20.12 inside Topaz Workbench. However, it will also be applicable to the rebranded BMC AMI DevX Total Test inside BMC AMI DevX Workbench for Eclipse.

## Prerequisites

- BMC AMI DevX Total Test 20.x: Fundamentals Concepts
- BMC AMI DevX Total Test 20.x: Fundamentals Preparing a Test Program

## Recommended Trainings

- Basic working knowledge of Topaz Workbench/BMC AMI DevX Workbench for Eclipse

## Learning Objectives

- Create a Total Test project
- Record tests from the Total Test and Xpediter perspectives
- Run a test scenario
- Describe the settings in the Execution Context dialog box
- Perform automated virtualized testing
  - Create a baseline test
  - Extend the baseline test
  - Test a new version of the program
  - Rebaseline the test
- Describe the communication architecture of a non-virtualized test
- Perform automated non-virtualized testing
  - Create a simple, functional test
  - Extend the test scenario
  - Prepare the test data using palette elements
  - Validate the test using palette elements

## Course Modules

### Module 1: Creating a Total Test Project

- Set Preferences
- Create a Project
- Best Practices

### Module 2: Recording Test Cases in Total Test

- Record a Test Case
- Create Stubs

### Module 3: Running Test Scenarios

- Execution Context Dialog Box
- Code Coverage

### Module 4: Virtualized Testing

- Create a Baseline Test
- Extend the Baseline Test
- Test the New Version of the Program
- Validate the Results
- Rebaseline the Test
- Best Practices

### Module 5: Non-virtualized Testing - Architecture and Repository

- Architectural Overview
- Repository in the Web client
- Defining Components in the Web Client
- Defining Components in the Eclipse Client

### Module 6: Non-virtualized Testing - Creating a Basic Test Scenario

- Create a Test Scenario in the Web Client
- Create a Test Scenario in the Eclipse Client
- Record the Test Scenario Using Xpediter

### Module 7: Non-virtualized Testing - Extending the Test Scenario

- Create the Test Scenario
- Extend the Test Scenario Using Iterate and Variable Elements

### Module 8: Non-virtualized Testing - Test Preparation

- Prepare the Test Using the SQL Select Element
- Use the Job Submit Element for Copying and Loading Data
- Use the Write Records Element

### Module 9: Non-virtualized Testing - Test Validation

- Validate the Db2 Data Using SQL Select and Verify Elements
- Validate the Dataset Content Using the Read Dataset element
- Validate the Job Output Using the Read Spool File Element
- Run Comparisons Using Palette Elements
- Best Practices

#### Discount Options

Have multiple students? Contact us to discuss hosting a private class for your organization.

Contact us for additional information 