

# TestComplete Support Plugin



# TestComplete

## About

A SmartBear plugin for running TestComplete tests from Jenkins.

The plugin provides a build step that lets you include your TestComplete tests in your Jenkins freestyle jobs and Pipelines. In addition, the plugin keeps a list of test runs and lets you view test results directly from Jenkins.

- [Plugin Information](#)
- [Usage](#)
- [More Information](#)
- [Technical Support](#)
- [Version History](#)

### Plugin Information

View TestComplete support [on the plugin site](#) for more information.

## Prepare Nodes

Prepare a test computer (node) for automated testing. Make sure the node has everything it needs to run tests successfully:

- TestComplete or TestExecute is installed on them.
- The required browsers and applications are installed on the nodes.
- TestComplete project files are copied to your node workspace.
- The required test data files and other helper files reside on the nodes.
- If you have a desktop tested application, make sure the application is installed on the nodes before running TestComplete tests.

To copy files to the nodes, you can, for example, use the Folder Copy operation of the [File Operations Plugin](#).

## Usage

- Freestyle Jobs
- Pipeline
- View Test Results

### *Freestyle Jobs*

To run your TestComplete tests as part of a Jenkins job:

1. Add the TestComplete Test step to your Jenkins job.
2. Configure the step:
  - Select the test runner: TestComplete or TestExecute.
  - Specify the TestComplete project suite file and select tests to be run.
  - Important: Select the **Run interactive user session** check box, if needed (see below), and specify additional parameters for the run.
3. Run the build as you normally would.

### *Pipeline*

To run your TestComplete tests as part of a Jenkins Pipeline:

1. Add code that will run your TestComplete tests to your Pipeline script (Jenkinsfile).
2. To generate the needed code, you can use the built-in Jenkins Snippet Generator utility:
  - Click the **Pipeline Syntax** link in the **Pipeline** section of your project.
  - In the **Sample Step** drop-down list, select testcompletetest: TestComplete Test.
  - Configure the step as described in the instruction for freestyle jobs above (select a test runner, specify a project suite and tests to run, and so on).
  - Click **Generate Pipeline Script**. Copy the generated code and then paste it to your Pipeline script.

3. Run your Pipeline as you normally would.

## View Test Results

After your build run is over, you can see the TestComplete Test Results link on the build page. Click the link to view test results. In the subsequent screen, click the link in the Test column to explore detailed test results.

## Important

Your test nodes must **run an interactive user session**. Otherwise, TestComplete (or TestExecute) will be unable to interact with the UI of the application under test.

To create an interactive user session:

- If you use TestComplete 10.6 or later, we recommend that you enable the **Run interactive user session** option of the TestComplete Test step (if you use Pipeline, set the useTCService parameter of the step to true).
- If you are unable to use this approach for some reason, then an alternative to running an interactive user session is to launch the node and control it via Java Web Start.

## More Information

You can find complete information on using the plugin in [TestComplete documentation](#).

## Technical Support

If you have any questions or need assistance with setting up the plugin and configuring the step's properties, please [contact the SmartBear Support Team](#).

## Version History

### Version 2.3

- *New feature:* Since TestComplete version 14.20, you can assign tags to scripts, keyword tests, BDD feature files and scenarios. Now, the plugin allows you to run tests by specifying tags or tag expressions.

### Version 2.2

- *New feature:* Support for the new test log format introduced in TestComplete version 14.1.

### Version 2.1

- *New feature:* Support for TestComplete 14 and TestExecute 14.
- *Fixed:* It was impossible to run TestComplete tests on multiple Jenkins nodes concurrently by using pipelines.
- *Fixed:* When running TestComplete tests using pipelines, the test results could incorrectly show the names of Jenkins nodes where the tests were run.

### Version 2.0

- *New feature:* Added support for Pipeline.

### Version 1.9

- *New feature:* Support for TestComplete/TestExecute x64. If you have the 64-bit version of TestComplete (or TestExecute) installed on the node, the plugin will use it to run tests.

### Version 1.8

- *New feature:* You can specify a screen resolution for interactive sessions in which the test step will run your TestComplete tests.

### Version 1.7

- *New feature:* Now you can use the test step to run tests in the CrossBrowserTesting cloud from your Jenkins nodes.
- *Fixed:* If TestComplete logs contained national characters, these characters were processed incorrectly in JUnit-style reports.

### Version 1.6

- *New feature:* Support for TestComplete 12 and TestExecute 12.

## Version 1.5

- *New feature:* Now you can specify custom command-line arguments to pass to TestComplete or TestExecute.
- *Fixed:* The plugin returned 0 as the exit code instead of the actual exit code that TestComplete or TestExecute returned.

## Version 1.4

- *New feature:* More detailed JUnit-style reports.
- A couple of minor bugs have been fixed.

## Version 1.3

- *New feature:* Support for TestComplete 11 and TestExecute 11.
- *Fixed:* It was impossible to run jobs with TestComplete Test steps on different nodes in parallel.
- *Fixed:* The TestComplete Jenkins plugin worked incorrectly if a slave node was controlled via a service started from a JNLP application.

## Version 1.2

- *New feature:* Generating test reports in the MHT format.
- *New feature:* Support for generating and publishing JUnit-style reports.
- *New feature:* Now you can use Jenkins variables to configure the TestComplete Test step.
- *Fixed:* An exception occurred when the TestComplete Test step was added to a Conditional Build Step.