

# Execute Tests

- [Overview](#)
- [Navigation and Actions bar](#)
  - [Export Automated Test Definition](#)
    - [Cucumber](#)
    - [Generic](#)
  - [Import Execution Results](#)
- [Execution Status](#)
- [Affected Requirements](#)
- [Comment](#)
- [Execution Defects](#)
- [Execution Evidence](#)
- [Issue Details](#)
  - [Test Description](#)
  - [Test Issue Links](#)
  - [Test Issue Attachments](#)
- [Pre-Conditions](#)
  - [Pre-Condition Attachments](#)
- [Custom Fields](#)
- [Test Details](#)
  - [Manual Test Step Results Table](#)
  - [Gherkin Test Details](#)
    - [Examples](#)
  - [Unstructured Test Details](#)
- [Automated Results](#)
- [Activity](#)
- [Attaching Screenshots](#)
- [Updating the definition of the Test after the execution started](#)
  - [Reset Execution](#)
  - [Merge Execution](#)

## Overview

You can either execute your tests manually or automatically. Even if it is an automated test, you can manually set the test status for that execution. You can also create custom statuses.

Tests are always executed within a Test Execution context, even if it's an ad hoc execution. You can execute tests inline (i.e., define the test run status), directly in the Test Execution issue screen, or you can use the Execution Screen which is detailed extensively below.

A Test Execution is validated if all the Tests that verify the specific environment target passed. In order to execute, monitor the progress, and report the results, each Test has its own Execution page details. Each Test associated with a Test Execution is called a **Test Run**. A Test Run is like an instance of the Test that contains a copy of the test definition (e.g., description, steps) and the result of that test in the context of that Test Execution.

When a Test definition is modified, the respective Test Runs aren't automatically updated. This happens because the execution of a Test depends directly on its definition. If the Test definition is changed, the executions might not be consistent with the definition anymore. Each time a Test Run is created, the Test definition is copied to it. This guarantees that when the Test definition changes, the Test Runs are not affected and will always contain the original (i.e., when it was added to the Test Execution) Test definition.

You are allowed to execute a Test if:

- you have permission to "**Browse**" the Test Execution issue
- you have permission to "**Resolve**" the Test Execution issue
- the Test Execution issue is not in a status configured to disallow executions
- the Test issue is not in a status configured to disallow executions



### Please note

Find more about how you can disallow executions based on the workflow status of Test or Test Execution issues at:

- [Miscellaneous](#)
- [Using Jira workflows for testing purposes](#)

Even though you might not have permission to execute a Test Run and if the execution has already been started, you can see the execution details page in **read-only** mode. An informative message will be shown to at the top of the execution page if the current Test Run is in read-only mode.

When you execute a Test, the Xray Execution Screen is shown. This screen is accessible from the Test Execution issue view page and the Test issue view page (Test Runs section), and this is where the execution results are displayed and registered. This page contains information about the Test definition and the Test Execution issue context to allow you to execute the Test without exiting this page.

The execution screen is composed of the following sections:

- [Navigation and Actions bar](#)
- [Execution Status and Test Execution fields](#)
- [Affected Requirements](#)
- [Comments](#)
- [Defects](#)
- [Evidence](#)
- [Test issue details](#)
- [Custom Fields](#)
- [Preconditions](#)
- [Test details \(script or steps\)](#)
- [Result contexts](#)
- [Activity](#)

Calculator / Test Execution: CALC-49 / Test: CALC-44

Square root of integers

Return to Test Execution

Next

Execution Status

PASS

Started On:

28/Aug/14 1:57 PM

Finished On:

29/Sep/14 10:21 PM

Assignee:

Administrator

Executed By:

Administrator

Tests environments:

Versions:

v1.0

Revision:

r12324

Comment

Preview Comment

This execution has failed with an error. Check the log file.

Execution Defects (0)

Create Defect

Create Sub-Task

Add Defects

No defects yet...

Execution Evidences (0)

Add Evidences

No evidences yet...

Execution Details

Test Description

This tests the square root operation of integers

Test Issue Links (2)

created

CALC-52

Error calculating the square root of integers

OPEN

tests

CALC-33

Square root operation

OPEN

Test Steps (3)

1

Step

I choose the **square root** operation on the calculator

Data

Expected Result

The operation must appear selected

Actual Result

Comment

Defects (0)

Evidence (0)

Step State

PASS

2

Step

I enter an integer into the calculator

Data

I = 9

Expected Result

Actual Result

Comment

Defects (0)

Evidence (0)

Step State

PASS

3

Step

I press the *Calculate* button

Data

Expected Result

The result 3 must be displayed in the screen, at the right of the "=" sign

Actual Result

Comment

Defects (0)

Evidence (0)

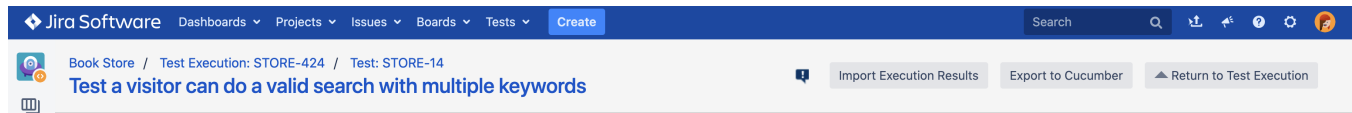
Step State

PASS

## Navigation and Actions bar

Always available, the navigation bar contains breadcrumbs to allow easy navigation between the Test issue hierarchy. It also contains a toolbar with the **Return to Test Execution** button that allows you to jump to the current Test Execution issue page, as well as the **Next** and **Previous** buttons, to quickly navigate to the execution screen of the other Tests included on the same Test Execution, if applicable.

On this bar, you can also find actions to **Export test definitions** and **Import execution results**. This is useful if you need to do an ad-hoc execution for an automated Test. Usually, automated tests are executed in a Continuous Integrating platform and the results are imported automatically into Xray (creating new Test Execution issues for the results) but you can also execute each automated Test individually and import the execution result.



## Export Automated Test Definition

If you are executing an automated Test, you can export its definition right from the execution screen.

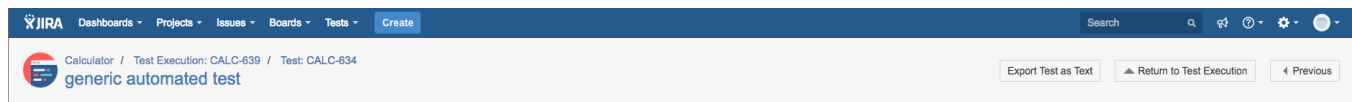
### Cucumber

If you're executing a Cucumber Test, an option to export the Test as a feature file will appear.



### Generic

If you're executing a Generic Test, then an option to export the Test as text will appear.



## Import Execution Results

If you already have the test results, just use this action to import the report files into Xray and update a specific Test Run.

## Execution Status

The **Execution Status** describes the current Test progress. The possible (native) status for a Test Run are **TODO**, **EXECUTING**, **FAIL**, **ABORTED**, and **PASS**. You can also create custom statuses in the Xray [Execute Tests](#) administration page.


You can change the status of the execution directly through this field. It supports inline editing and once activated, it will display a set of statuses that you can set the Test Run. If the option for setting the Test Run status manually without having to execute all Test Steps (or Examples, as in the case of Cucumber Tests) is enabled in the Xray administration page, then you can set the Test Run Status to any status, independent of the current aggregate status of the Test Steps (or Examples). If this option is disabled, then the status that you can set through this field will be limited to the available transitions based on the current Steps or Examples aggregated statuses.


The status section contains other relevant fields such as:


- **Assignee** - the User assigned to perform the current test execution
- **Executed By** - the last User that changed the status of the current test run
- **Versions** - the target release version tested by the current test execution
- **Revision** - the source code and documentation version used in the current test execution
- **Started On** - the date and time the execution of the current test started
- **Finished On** - the date and time the execution of the current test finished



You can change the Test Run assignee directly on the field. It supports inline editing and once activated, you can filter by the user and then select the one you wish to assign to the Test Run.


Assignee: **Administrator**  Versions: **v1.0**

Executed By:  **Administrator**  
 admin@xpand-it.com


Tests environments:  11:44 AM - Tuesday - London

Activity More




---


Assignee: administrator 

Executed By: **Showing 1 of 1 matching users**


Tests environments:  **Administrator - admin...**

If the dates are configured to be edited, you can edit the Started On date by clicking in the calendar icon and choosing the date and time. You can check your configurations by going to Jira Administration > Add-ons > Global Preferences.

Execution Status  **PASS**  

Started On: 15/Nov/17 09:24 PM  Finished On: 16/Nov/17 9:24 AM

Affected Requirements

 WEB-818 As a user i can login with my Facebook

Comment

*Click to add comment*

Execution Defects (0)

No

**Note:** The Jira `jira.date.time.picker.java.format` and `jira.date.time.picker.javascript.format` properties in Jira Administration advanced settings should belong to the same pair of properties in order for the datetime picker work properly. See the following link for more information

- <https://confluence.atlassian.com/jira/changing-the-due-date-input-format-192536.html>

## Affected Requirements

This section provides the ability to manually set the status for Requirement issues that are tested by the current Test issue. By default, the requirement status is calculated based on the latest Test Run of each Test associated with a given requirement. Even though the Test Run status is **FAILED**, not all requirement issues associated with the Test issue might have failed. Some of these requirements can, for instance, be **PASSED**, if the tester chooses to explicitly set the requirement status. This functionality makes it possible for a single Test issue to test multiple requirement issues with different concerns and functionalities. These requirement statuses that are explicitly set in a Test Run will then be considered when calculating the Requirement Status and Requirement Coverage.

 This section is only visible if the "Separation of Concerns" option is disabled in [Coverage Settings](#) settings page.

Affected Requirements

☒ CALC-33 Square root operation OPEN OK Not Run

Comment Preview Comment

This execution has failed with an error! Check the log file.

Execution Defects (0) Create Defect Create Sub-Task Add Defects

Execution Evidences (0) Add Evidences

All affected requirement statuses will be updated whenever a step (or example in the case of Cucumber automated Tests) status is changed. The status for any affected requirement can always be changed by clicking on the status and choosing the desired option.

## Comment

The Comment field is inline editable to input overall appreciations about the current execution without reloading the page

The field supports the [Jira markup wiki language](#).

Comment

Preview Comment ^

This execution has failed with an error! Check the log file.


Click to edit

## Execution Defects

The Execution Defects field lists the overall defects related to the current execution. It is possible to create a new Defect issue, create a new Sub-task or Add existing Defects by clicking on the respective option.

Execution Defects (0)

Create Defect | Create Sub-Task | Add Defects ^



No defects yet...

## Execution Evidence

The Execution Evidences field lists the overall attachments related to the current execution and by clicking on the **Add Evidences**, opens the **Attach Files** dialog box.

Execution Evidences (0)

Add Evidences ^



No evidences yet...

## Issue Details

This section includes some fields present on the Test issue. It often contains useful information for testers to execute the Test.

## Test Description

This section includes the current Test issue description. The Test issue description often contains useful information for testers to execute the Test.

The field supports the [Jira markup wiki language](#).

▶ Execution Details

Test Description





This tests the square root operation of integers

## Test Issue Links

This section shows all links associated with the Test.

Test Issue Links (2)

^


|         |   |  |
|---------|---|--|
| created |  <a href="#">CALC-52</a> Error calculating the square root of integers |  <a href="#">OPEN</a> |
| tests   |  <a href="#">CALC-33</a> Square root operation                         |  <a href="#">OPEN</a> |

## Test Issue Attachments

This section shows all attachments associated with the Test.

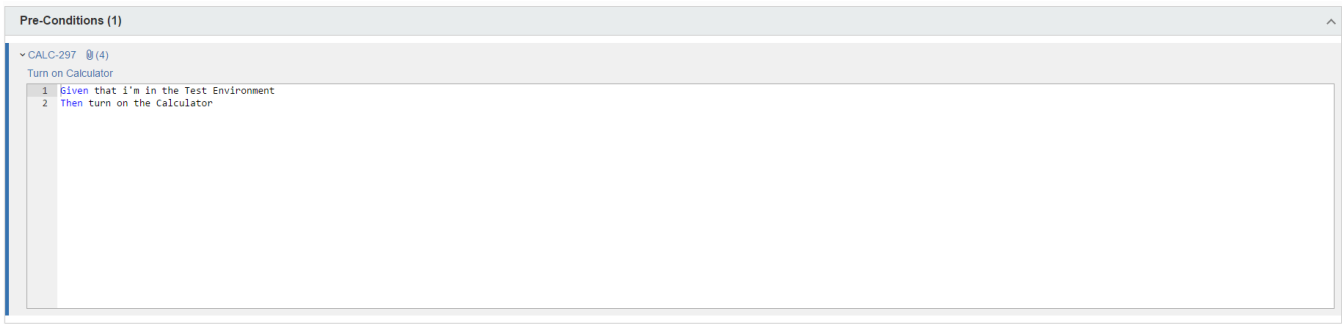
Test Issue Attachments (1)

^

 [ForTest.json](#) 47 kB Today 5:26 PM

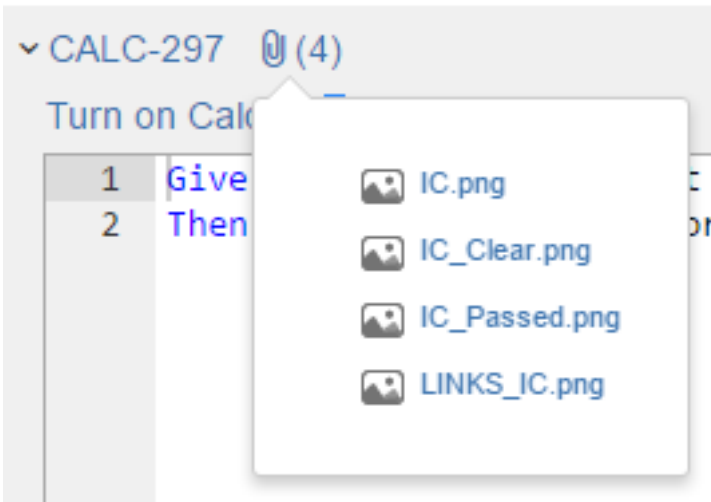
## Pre-Conditions

The Pre-Condition field shows the Pre-Conditions issues associated with the current Test.



## Pre-Condition Attachments

The Pre-Condition field also shows the Pre-Conditions attachment in a inline dialog.



## Custom Fields

You define additional custom fields for Test Runs. These fields can be useful to add extra information to Test Runs, usually only available when or after executing Tests.

Test Run custom fields can be configured by project and by Test Type. Therefore, these settings will not affect other projects within your Jira instance. For instance, it is possible to have custom fields just for Manual Tests within a project.

If there are any custom fields defined for the scope of the Test Run, they will appear on the "**Custom Fields**" section.

| Custom Fields      |                     |             |
|--------------------|---------------------|-------------|
| Avg CPU usage*     | Requests per second | Error rate* |
| Negligent 0% - 20% | 45                  | 0.02        |

It is also possible to define custom fields that are **mandatory**. If this is the case, users will not be able to set a final status on the Test Run if there is any required custom field that is not populated.

## Load testing subscription

Execution Status  **TODO**

Started On:  Finished On: 

Final status transitions are disabled, Test Run is missing required fields.

**Comment** Preview Comment 

*Click to add comment*

**Execution Defects (0)** Cre

## Test Details

Within the Test Details section, you will find the Test script to execute. Depending on the Test Type, this section can display Test Steps, a Generic Definition field (for unstructured Tests), and Gherkin scenarios.

## Manual Test Step Results Table

A **Manual Test** execution screen contains the **Test Steps** field, listing the Steps, Data, Expected Result and Attachments to be used as reference while testing.

By clicking on each step, the following fields become editable:

- the **Actual Result** field which contains the actual result of the execution of the step;
  - This field can be edited using the wiki markup toolbar from Jira
  - It's possible to refer to an Evidence previously added to any Step or Global execution
- the **Comment** field which contains any notes about the corresponding step;
- the **Defects** field which contains any defect associated with the corresponding step. It is also possible to input new defects by:
  - Clicking on the Create Defect to create a new Defect
  - Clicking on the Create Sub-Task to create a new Sub-Task
  - Clicking on the Add Defects to add an already existing defect
- the **Evidences** field which contains any attachment associated with the corresponding step. It is also possible to input new attachments by clicking on the + button, which opens the **Attach Files** dialogue box
- the **Status** which contains the current step testing status. This can be changed by clicking on any of the status in front of your Step Status.

Note: Changing the Status on the each Step will affect the **Execution Status** field on the Execution Screen.



Test Steps (3)

1

Step

I choose the **square root** operation on the calculator

Data

Expected Result

The operation must appear selected

Actual Result

Comment

Defects (0)

Evidence (0)

Step State **PASS**

2

Step

I enter an integer into the calculator

Data

I = 9

Expected Result

Actual Result

Comment

Defects (0)

Evidence (0)

Step State **PASS**

3

Step

I press the Calculate button

Data

Expected Result

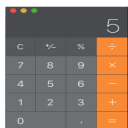
The result 3 must be displayed in the screen, at the right of the "=" sign

Actual Result

Style

B I U A

The result was 5. Please see attachment:



Visual Text

Save Cancel

Actual Result

Comment

Defects (0)

Evidence (2)

Step State **FAIL**

## Gherkin Test Details

Gherkin Tests always display the Test scenario on the execution page. Usually, these tests are automated and their result can be imported into Xray. However, it is also possible to execute these Tests manually by setting the overall result of the Test Run, or by setting the result of each example (in case of scenario outlines).

Test Details

Test Type:

Automated[Cucumber]

Scenario Type:

Scenario Outline

Scenario:

1

Given I have entered <input\_1> into the calculator

2

And I have entered <input\_2> into the calculator

3

When I press <button>

4

Then the result should be <output> on the screen

5

6

Examples:

7

| input_1 | input_2 | button | output |
|---------|---------|--------|--------|
| 30      | 20      | sub    | 10     |
| 5       | 2       | sub    | 3      |
| 40      | 0       | sub    | 0      |

8

9

10

## Examples

A scenario **outline** of an automated Cucumber test will normally contain an "examples" table containing several inputs and outputs for the given scenario.

The examples section details these examples, containing an overall status bar in the header and a table with all the examples below. Each example can have the status: PASS, FAIL or TODO.

A scenario outline test is **PASS** only if all examples passed.

A scenario outline test is **FAIL** if at least one of the examples failed.

A scenario outline test is **TODO** if at least one of the examples did not run and all the other examples passed or todo.

Examples

| <input_1> | <input_2> | <button> | <output> | Status      |
|-----------|-----------|----------|----------|-------------|
| 30        | 20        | sub      | 10       | <b>PASS</b> |
| 5         | 2         | sub      | 3        | <b>FAIL</b> |
| 40        | 0         | sub      | 0        | <b>TODO</b> |

## Unstructured Test Details

Unstructured Tests only contain the generic **definition field** on the Test Details section. This field can specify a test script, an ID to an external (automated) test case, or an Exploratory Test charter. Similar to Gherkin Tests, most unstructured test results will be imported or set by another tool. However, it is also possible to set the status of unstructured tests manually.

| Test Details |  | ^ |
|--------------|--|---|
| Test Type:   | Generic                                  |   |
| Definition:  | jmeter.bookstore.newsletter.subscription |   |

## Automated Results

Xray for Jira supports the import of multiple execution results for the same Test Run. Multiple executions for the same Test Run specifies different results for different contexts/environments. Multiple executions are only supported for Automated Tests. Manual Tests do not have this concept.

When a Test Run contains multiple executions imported from external sources like [Calabash](#) and [Xamarin Test Cloud](#), these executions are presented in a "Results" section on the execution page.

This section contains a Table with all executions for the same Test Run. The table has the following columns:

- **Context:** An execution context name (in the case of Xamarin Test Cloud results, the mobile device, and operating system version)
- **Error Message:** A possible error message if the result has failed.
- **Duration:** The complete duration of an execution result.
- **Status:** The execution results status.

▼ Results

| Context                                   | Error Message |          | Duration    | Status |
|---|---------------|----------|-------------|--------|
| ▼ HTC One (4.3)                           |               |          | 16 millisec | PASS   |
| <input_1>                                 | <input_2>     | <button> | <output>    | Status |
| 20  | 30            | add      | 50          | PASS   |
| 2   | 5             | add      | 7           | PASS   |
| 0   | 40            | add      | 40          | PASS   |
| ▶ Samsung Galaxy S III Mini (NFC) (4.1.2) |               |          | 16 millisec | PASS   |

## Activity

The Activity field lists all the activities related to the current execution.

▼ Activity



admin changed the comment of this execution.

New Value

While the Test Execution was being performed, a recurring Bug was detected, represented on the Evidence 3.jpg.

Today 10:31 PM



admin added attachments to this execution:

- Evidence 3.jpg

Today 10:24 PM

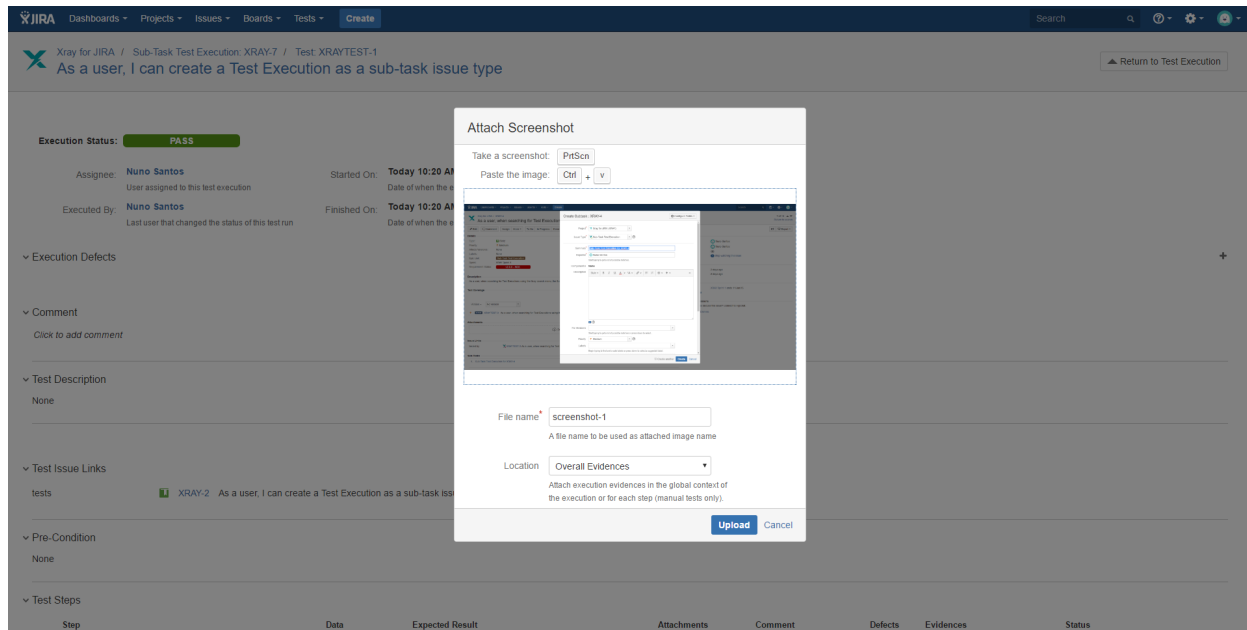


admin associated defect "DEMO-36-Bug 5 - Trivial" with this execution.

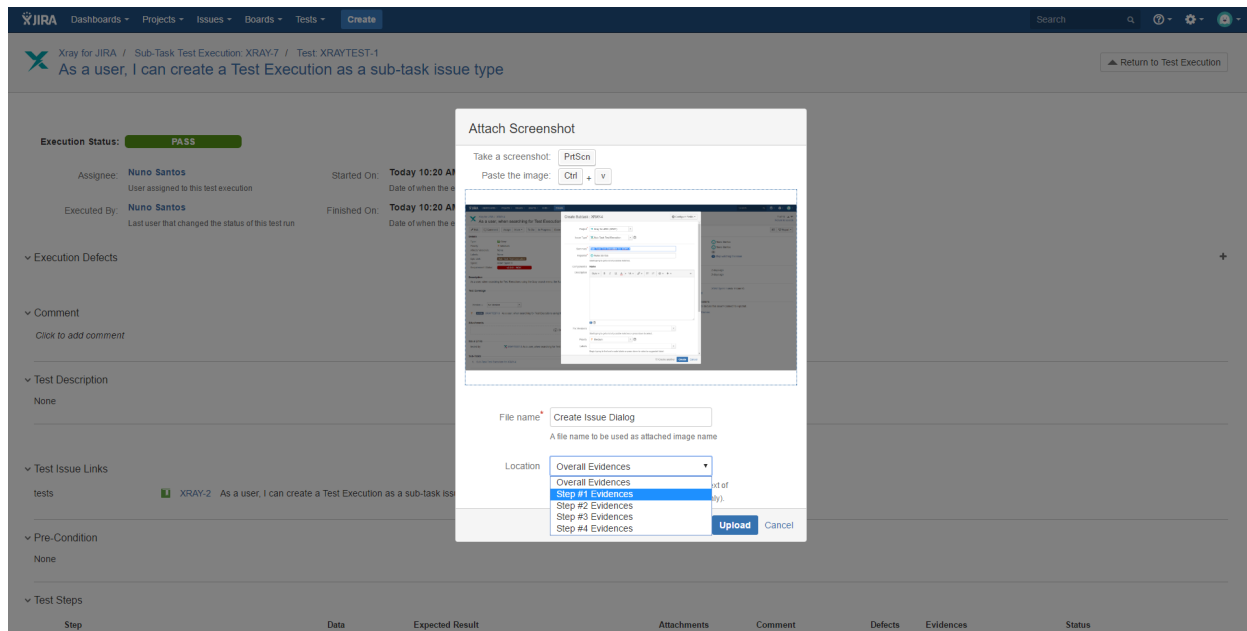
Today 10:23 PM

# Attaching Screenshots

To attach a screenshot, paste the image from your clipboard using the relevant keyboard shortcut. (For Mac OSX, use **CMD+V**; for Windows, use **Ctrl+V**.)



Enter a file name for the screenshot you are attaching (defaults to "screenshot-" concatenated with the next number available) and choose where you want to place the evidence (Overall Execution or Test Steps).



Click **Upload** to attach the captured image to your Test Run.

## Updating the definition of the Test after the execution started

When a Test definition has changed, a new dialog box will appear on the top of the Execution page informing you that the current execution is obsolete, and will prompt you with Reset or Merge buttons that, when pressed, will load the updated definition as well as delete all the execution data or merge the new Test definition into the current state of execution.



Test definition has been modified. You can reset the execution of this test by pressing the "Reset" button. This will load the new test definition and delete the current execution data. If you wish to update the Test definition and maintain the current execution, you can click on the "Merge" button.

Reset

Merge

## Reset Execution

You can choose to reset the Execution or keep it for historic record. If you reset the execution, all information previously saved (status, evidence, defects) will be erased.

## Merge Execution

You can choose to merge the new Test definition into the current status of execution. This will keep steps that weren't changed along with the associated results/comments/evidence. It will also load new steps (moved, erased, or created), pre-conditions, or attachments definitions. This will always force a recalculation of Test Run Status. For instance, if the current Test Run Status is FAIL because of a step and this step is removed from Test, if you merge, then the Test Run Status will change to whatever the value combination of the others steps returns.