

# GA Release Notes

## June 2018

The Xray team is proud to announce the release of **Xray Cloud 1.1.5-1.001.000**.

This initial version enables you to:

- create manual and automated Tests
- implement BDD through native Cucumber support
- organize Tests into Test Sets
- create a Test Plan by defining the scope for a test campaign (e.g., a specific version or sprint) and check the overall testing progress
- schedule Test Executions
- schedule and manage executions for Test Environments
- conduct coverage analysis by version, Test Plan, Test Environment
- ready for automation, by providing a built-in REST API for submission of test results
- generate several reports, including those for Test Coverage analysis

Discover the key features below 😊

- [Release highlights](#)
  - [Tests](#)
  - [Preconditions](#)
  - [Test Sets](#)
  - [Test Executions](#)
  - [Execute Test Runs](#)
  - [Test Plans](#)
  - [Reports](#)
  - [Test Environments](#)
  - [Cucumber](#)
  - [REST API](#)
  - [Project-Level Settings](#)

## Release highlights

# Tests

In very simple terms, a Test is a sequence of steps coupled with conditions or variables, test inputs and an expected result. It is intended to establish the quality, performance or reliability of a component of a system, i.e., test target. Usually, every requirement or objective the test target is expected to achieve needs at least one Test. The success of the Test is determined by comparing the expected and actual results. A Test is an issue type installed by Xray that can be executed multiple times.

Learn more about Tests [here](#).

### Add two numbers

EditCommentAssignTo DoIn ProgressDoneAdmin

Type:TestStatus:TO DO

Priority:Medium

Resolution:Unresolved

Affects Version/s:None

Fix Version/s:None

Component/s:None

Labels:None

Description

Click to add description

Test Details

TypeCucumber

Scenario

1Given I have entered <input\_1> into the calculator

2And I have entered <input\_2> into the calculator

3When I press <button>

4Then the result should be <output> on the screen

5

6Examples:

7| input\_1 | input\_2 | button | output |

8| 20 | 30 | add | 50 |

9| 2 | 5 | add | 7 |

10| 0 | 40 | add | 40 |

11| 1 | 40 | add | 41 |

### Subtract two numbers

EditCommentAssignTo DoIn ProgressDoneAdmin

Type:TestStatus:TO DO

Priority:Medium

Resolution:Unresolved

Affects Version/s:None

Fix Version/s:None

Component/s:None

Labels:None

Description

Click to add description

Test Details


TypeManual


Manual Steps

Action	Data	Expected Result	Attachments
1 I choose the operation of the calculator Subtraction		The operation must appear selected.	
2 I enter the input into the calculator	I1: 5 I2: 2		
3 I press the Calculate button		The result 3 should be displayed in the screen, on the right of the "=" sign.	

# Preconditions

## Turn on the calculator

 Edit

 Comment


Assign


To Do

In Progress

Done

Admin ▾

Type:  Precondition Status: **TO DO** [\(View workflow\)](#)

Priority:  Medium Resolution: Unresolved

Affects Version/s: None Fix Version/s: None

Component/s: None

Labels: None

### Description

*Click to add description*

### Precondition Details

Type

Cucumber

Scenario

- 1
- Given that I'm in the test environment
- 2
- Then the calculator must be turned on

Preconditions specify the conditions that need to be fulfilled before executing a test.

A Precondition is like defining the step "0" for your tests. This is convenient when you have many use cases that start by doing exactly the same thing. For example, if you are going to do something on a system, you have to do several things (e.g., restore the DB, authenticate) before going through your Test steps. Each of those things may be done and used independently, i.e., some Tests may require just restoring the DB, another may require just the authentication, others may require both or more.

Learn more [here](#).




# Test Sets


A Test Set is a collection of Tests. It often contains detailed instructions or goals as well as information on the configuration to be used during testing.

Test Sets are simple, flat lists of Tests that you can use as a basis for creating Test Executions or Test Plans. Since the Tests are grouped in some logical way, such as grouping all Tests related with regression testing or security, or all Tests for a specific component or some high-level feature/business case, a Test can be part of different Test Sets. Test Sets can also be used as a dynamic way to cover requirements.


Learn more [here](#).

## Calculator operations

-  Edit
-  Comment
- Assign
- To Do
- In Progress
- Done
- Admin 

Type:  Test Set

Status: **TO DO** [\(View workflow\)](#)

Priority:  Medium

Resolution: Unresolved

Affects Version/s: None

Fix Version/s: None

Component/s: None

Labels: None



Description


[Click to add description](#)


### Tests

Create Test

Add Tests

10 

Columns 

	Key	Summary	Status	Actions
<input type="checkbox"/>	<a href="#">CALC-1</a>	Add two numbers	TO DO	...
<input type="checkbox"/>	<a href="#">CALC-2</a>	Subtract two numbers	TO DO	...
<input type="checkbox"/>	<a href="#">CALC-3</a>	Divide two numbers	TO DO	...
<input type="checkbox"/>	<a href="#">CALC-4</a>	Multiply two numbers	TO DO	...

Prev **1** Next

Total **4** issues

## Test Executions

A Test Execution is an issue type that aggregates a user-determined collection of Tests. It monitors and verifies if those Tests are working as expected in a target context and environment. Through the Overall Execution Status, updated as each test is being performed, the user is informed about the progress of the Test Execution, including which tests passed, failed, are being executed or waiting to be performed.

An association between a Test Execution issue and a Test issue is called a **Test Run**. Hence, a Test Execution is composed of a list of Test Runs.

A Test Execution issue can be assigned to a Tester. It is also possible to assign individual Test Runs to several Testers.

Learn more [here](#).

Assert calculator basic operations are returning expected results

EditCommentAssignTo DoIn ProgressDoneAdmin

Type:Test ExecutionStatus:TO DO (View workflow)Priority:MediumAffects Version/s:NoneComponent/s:SimpleLabels:NoneRevision:u2yt34fwu4y37tdgxu2365cuw

DescriptionClick to add description

TestsCreate TestAdd

Overall Execution StatusTOTAL TESTS: 43 PASSED1 FAILED

Filters

10Columns

Key	Summary	Test Type	Status	Actions
<input type="checkbox"/> CALC-1	Add two numbers	Cucumber	PASSED	⋮
<input type="checkbox"/> CALC-2	Subtract two numbers	Manual	FAILED	⋮
<input type="checkbox"/> CALC-3	Divide two numbers	Cucumber	PASSED	⋮
<input type="checkbox"/> CALC-4	Multiply two numbers	Manual	PASSED	⋮

Test PlansAssociated Test PlansCALC-9

Test EnvironmentsAssociated Test EnvironmentsAndroid

Assignee: Bruno CondeReporter: Bruno CondeVotes: 0Watchers: Stop watching this issueCreated: 4 minutes agoUpdated: 4 minutes agoBegin Date: 30/Jul/18 11:52 AMEnd Date: 30/Aug/18 11:52 AM

AgileView on Board

HipChat discussionsDo you want to discuss this issue? Connect to HipChat.ConnectDismiss

# Execute Test Runs

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 web panel). This is where the execution results are displayed and registered. It contains information about the Test definition and the Test Execution issue context to allow you to execute the Test without exiting this page.

Learn more [here](#).

Calculator / Test Execution: CALC-13 / Test: CALC-2

Import Execution Results

Return to Test Execution

Previous

Next

Subtract two numbers

Execution Status **FAILED**

Assignee: Bruno Conde

Executed By: Bruno Conde

Test Environments: Android

Started On: 29/May/2018 12:08 PM

Finished On: 29/May/2018 12:09 PM

Versions: 1.0

Revision: -

Comment

Preview comment

Execution Defects (1)

Add Defects

Create Defect

Execution Evidences (0)

Add Evidences

Execution Details

Test Description

This is the description for this Test

Test Issue Links (1)

tests

CALC-11 Addition Operation

↑

TO DO

calculator.png

395.5 kB 25/May/2018 2:02 PM

Test Steps (3)

1

Action

Attachments (1)

I choose the operation of the calculator **Subtraction**

Data

Expected Result

The operation must appear selected.

Comment

Defects

Evidences

Step State PASSED

2

Action

Data

Expected Result

Test Steps (3)

1

Action

Attachments (1)

I choose the operation of the calculator **Subtraction**

Data

Expected Result

The operation must appear selected.

Comment

Defects

Evidences

Step State PASSED

2

Action

Data

Expected Result

I enter the input into the calculator

It: 5

It: 2

Comment

Defects

Evidences

Step State PASSED

3

Action

Data

Expected Result

I press the **Calculate** button

The result **3** should be displayed in the screen, on the right of the "=" sign.

Comment

Defects

Evidences

Step State FAILED

CALC-15 Error while subtracting 2 numbers

# Test Plans

Tests

All Environments, final status

Create Test Execution

Add

Overall Execution Status

TOTAL TESTS: 6

2 PASSED

3 FAILED

1 TO DO

Filters

10

Columns

	Key	Summary	Assignee	#Test Executions	Latest Status		Actions
<input type="checkbox"/>	CALC-1	Add two numbers	Bruno Conde	2	FAILED		...
<input type="checkbox"/>	CALC-2	Subtract two numbers	Bruno Conde	2	FAILED		...
<input type="checkbox"/>	CALC-3	Divide two numbers	Bruno Conde	2	PASSED		...
<input type="checkbox"/>	CALC-4	Multiply two numbers	Bruno Conde	2	PASSED		...
<input type="checkbox"/>	CALC-5	Multiple Operations	Bruno Conde	1	FAILED		...
<input type="checkbox"/>	CALC-7	Test Scientific Mode	Bruno Conde	1	TO DO		...

Prev

1

Next

Total 6 issues

The purpose of a Test Plan is to enable you to have a better visibility and management over your Tests. It provides a table that shows all the Tests added to the Test Plan and their respective Test Executions that are also associated with the Test Plan. If tracking a particular version, the Test Plan will present the latest status for each Test, independent of the number of testing iterations (i.e., Test Executions) you make with them. You can group a bunch of Test Executions and easily give feedback on the current/latest status of the Tests involved in those Test Executions.

From the Test Plan, it's possible to create one or more Test Executions, for all or just a subset of the Tests contained within the Test Plan (e.g., only those Tests that are still failing). You are also able to assign one or more Test Plans to a given release and track progress of those tests within each Test Plan.

Learn more [here](#).

# Reports

Xray provides several reports to quickly evaluate the quality of your project, the testing progress and the status of requirements/stories being addressed in each sprint or in each version.

Learn more [here](#).



### Test Coverage

Calculates the test coverage status for a list of covered issues, in a normalised horizontal stacked bar chart, for the selected version, or Test Plan, and/or Test Environment.



### Test Executions List

Shows a list of Test Executions, their attributes and information about overall progress and linked defects.



### Test Plans List

Shows a list of Test Plans, their attributes and information about overall progress.



### Test Sets List

Shows a list of Test Sets and their attributes, along with an overview of the calculated status for the Tests that are part of the Test Set, for the selected version or Test Plan, and/or Test Environment.



### Tests List

Shows a list of Tests and their attributes, along with the calculated status for the Tests, for the selected version or Test Plan, and/or Test Environment.



## Test Environments

### Test Runs of Test CALC-1

CALC-1

Add two numbers

Latest Status

 **FAILED**

▾

All Environments, final status ▾

10 ▾

Columns ▾

Key ▾	Summary ▾	Test Environment ▾	Status ▾	Action	
<input type="checkbox"/> <a href="#">CALC-13</a>	Assert calculator basic operations are returning expected results	ANDROID	<div><div></div><div>PASSED</div></div>		
<input type="checkbox"/> <a href="#">CALC-12</a>	Ad-hoc execution for CALC-1	IOS	<div><div></div><div>FAILED</div></div>		
<input type="checkbox"/> <a href="#">CALC-10</a>	Test Execution for Test Plan CALC-9	ANDROID	<div><div></div><div>FAILED</div></div>		

Prev

1

Next

Total 3 issues

The status of a Test is generally calculated by looking at the Test's last execution. However, this does not work well if you execute the same Test in two different test environments (e.g., devices) and you want the two results to be consolidated.

Within a Test Execution, you may specify the **Test Environment(s)** where the tests will be executed. Test Environments can be created by project managers and made accessible for users to specify in Test Execution issues.

If you use Test Environments, you may execute the same Test for multiple test environments by creating Test Executions for each one. If you don't use Test Environments and you want to track tests for multiple environments (e.g., devices), then the only way to do it is to create multiple tests, one per each test environment.

Let's say that you have executions for two Test Environments: "Android" and "iOS". The test is considered as PASS only if the **latest** executions for Android and iOS are **both** PASS; otherwise, it will be FAIL. Xray allows you to calculate the status of Tests and Requirements considering all Environments or a specific Environment.

Learn more [here](#).

## Cucumber

Cucumber is a framework for BDD that enables clear understanding of features being implemented and their validation scenarios / acceptance criteria.

With Xray, you can make and manage the specification of Cucumber Tests (i.e., Scenarios / Scenario Outlines) as well as Cucumber Preconditions (i.e., Backgrounds).

The specification can be extracted from Xray and then it can be run in the CI tool against the code that implements each Gherkin phrase. Finally, the results can be submitted back to Xray either by using the UI or the REST API.

Learn more [here](#).



### 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
```

#### Examples:

	input_1	input_2	button	output
	20	30	add	50
	2	5	add	7
	0	40	add	40
	1	40	add	41

## REST API

Xray provides a REST API that enables you to import results from Cucumber automated tests as well as Manual and Generic Tests by using specific endpoints.

Learn more [here](#).



## Project-Level Settings

Xray provides project- and global-level settings. Project settings can be configured by project administrators. Some settings can be inherited from global settings; some are only global and some only belong to the project level.

Learn more [here](#).  
[blocked URL](#)