Xray 5.0.0 Release Notes

2 July, 2021

The Xray team is proud to announce the release of **Xray 5.0**.

This version's major features include **parameterized tests**, **data-driven testing**, **support for Robot Framework v4.0,** and **grouping defects and evidence** on the overall panel within the execution screen.

est version

- Upgrade Notes
  - Upgrading Xray to 5
  - Downgrading from Xray 5
    - Historical Coverage Gadget
- Release highlights
  - Parameterized tests
  - Group defects and evidence on the execution screen
  - Search Test Steps
  - Support for Robot Framework 4.0
- And more...
  - Document Generator
- Just one more, yet important, thing...
- All New Features and Bug Fixes in this release

Upgrade Notes

Upgrading Xray to 5

If you are upgrading from a version lower than 3.2, this implies that an upgrade task will run to pre-calculate historical data. This is a one-time action that will run in the background so that Xray is still available during the upgrade. A **re-index** operation of your Jira instance is required so that the internal data becomes searchable and usable by Xray.

⚠️ Please be aware that this upgrade task will run immediately after updating Xray. As this operation can take some time and use a considerable amount of server resources, we suggest scheduling the upgrade of Xray outside of business hours. It is always possible to cancel the recalculation operation in the Xray configuration section Custom Fields Maintenance.

If you are upgrading from v3.2.x or v3.3.x to v5, you must recalculate data in the Xray configuration section Custom Fields. This recalculation is necessary because of the huge improvement in the amount of data that Xray needs to add to the table and to the Lucene indexes.

⚠️ As this operation can take some time and use a considerable amount of server resources, we suggest scheduling the upgrade of Xray and the recalculation of the custom fields, outside of business hours.

Downgrading from Xray 5

Downgrading from Xray 5.x.x to a previous version will have the following side-effects on Test Runs with a dataset:

- Test parameters will no longer be converted on the Test Run page, this includes Test Step definitions and Pre-Conditions;
- When loading a Test Run created on version 5.x.x, the merge/reset message will be displayed;
- When merging a Test Run created on version 5.x.x with the new Test Step definition, all Test Step results will be updated but the reset/merge message will still be displayed;
- A Test Run created on version 5.x.x with iterations will be displayed in a "flat mode." The iterations will no longer be available and all the Test Step results from each iteration will be presented on the Test Steps section;

- After the downgrade, the integrity checker will fail on "Check for invalid Mementos" for all Test Runs created on version 5.x.x. We recommend running the integrity checker before downgrading to get rid of possible invalid mementos.

The Test Step Custom Fields and Test Run Custom Fields of type User Picker will have to be deleted in the project configuration as they are not supported. Following that, the Test Runs that are using a User Picker field will have to be merged or reset. This includes Test Runs that have the field for the Test Run itself and also Test Runs for Manual Tests where the Test Steps contained the User Picker field. Downgrading from Xray 5.x to a version less than 3.2.0 implies that you must perform a re-index afterward. This is because from Xray 3.2.0 onwards, the content and format of what Xray indexes have changed.

Historical Coverage Gadget

Some configuration parameters for this gadget were modified as a result of supporting multiple projects whenever using a saved filter as a source. It is possible that some users will get errors on the gadget after downgrading from 4.x. If this happens, please edit the configuration to ensure the gadget is configured properly.

Release highlights

Parameterized tests 🙂

Test parameterization is a powerful practice that allows the same test to be executed multiple times with different parameters. Parameters are similar to input values (variables) that can change with each execution.

Parameterized tests in Xray are defined just like any other test with the addition of some parameter names within the specification using the following notation: **${PARAMETER_ NAME}.** This notation is used to reference parameters within the **test steps**.

Precondition issues can also be parameterized by including parameter names in the precondition specification.

The parameters, along with their values, are defined within a **dataset**. A dataset is a collection of data represented with a tabular view where every column of the table represents a particular variable (or **parameter**), and each row corresponds to a given record (or **iteration**) of the dataset. The number of rows in the dataset determines the number of iterations to execute.

A dataset can be defined in the following entities/scopes:

1. Test (default dataset)
2. Test Plan - Test
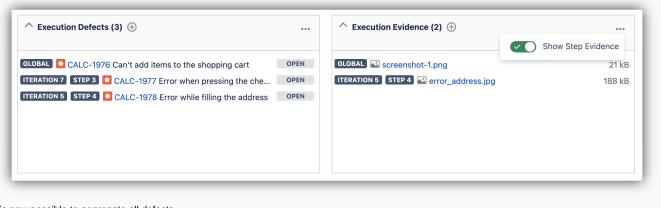3. Test Execution - Test (Test Run)

The closest dataset to the test run will be the one used to generate the iterations, effectively overriding any dataset defined in higher levels.

All iterations for a given test are executed within the context of the same test run. Each iteration can be expanded, and the steps executed individually. The step parameters will be replaced by the corresponding iteration values. The steps affect the iteration status, which in turn affects the overall test run status.

Learn more here.

Dataset for Test CALC-1974

There are a total of 24 iteration(s) to execute.

Create parameter   Import

Combinatorial Parameters

Gift
Yes
No

Quantity
1
2
5
10

#	Item	Price	Rating
1	In Search of Lost Time	$34	5
2	One Hundred Years of Soli...	$20	4.9
3	The Great Gatsby	$39	4.7

New

Type: Manual   Dataset

Manual Steps:

Steps dialog   Grid   Import   Export   Add Step

1   Action*

Style   B   I   U   A   A   &   &   ≡   ≡   ☺   + ${

Add ${Quantity} books of ${

Item
Price
Rating
Author
Gift

After items are added, a confirmation message appears mentioning ${Quantity} items of ${Item} were added to the basket.

Attachments (0)

Done   Cancel

Test Details   MANUAL

Custom Fields

Test Description

Iterations 24

Iteration 1 -   In Search of Lost Time $34   5   Marcel Proust   Yes   1   PASS

Iteration 2 -   One Hundred Years of ... $20   4.9   Gabriel Garcia Marquez   Yes   1   PASS

Iteration 3 -   The Great Gatsby $39   4.7   F. Scott Fitzgerald   Yes   1   PASS

Iteration 4 -   In Search of Lost Time $34   5   Marcel Proust   No   1   PASS

Iteration 5 -   One Hundred Years of ... $20   4.9   Gabriel Garcia Marquez   No   1   PASS

Iteration 6 -   The Great Gatsby $39   4.7   F. Scott Fitzgerald   No   1   PASS

Iteration 7 -   In Search of Lost Time $34   5   Marcel Proust   Yes   2   EXECUTING

Test Steps 11

1	Action	Data	Expected Result
	Add 2 books of In Search of Lost Time	None	After items are added, a confirmation message appears mentioning 2 items of In Search of Lost Time were added to the basket.

Actual Result   Comment   Defects (0)   Evidence (0)   Step State   PASS

2	Action	Data	Expected Result
	Click on the basket icon located on the top right toolbar of the app.	None	The basket page is displayed containing all 5 items.

Actual Result   Comment   Defects (0)   Evidence (0)   Step State   PASS

3	Action	Data	Expected Result
	Press the Checkout button to start the checkout process.	None	The checkout process is initiated asking the user the address details.

Actual Result   Comment   Defects (0)   Evidence (0)   Step State   TODO

Group defects and evidence on the execution screen

It is now possible to aggregate all defects and evidence on the overall top panel within the execution screen. This way, both the global defects and the defects associated with steps will be shown on the same panel, making it easier to view all the defects created during execution. The same is also true for evidence.

This feature is optional and you can disable it within the execution screen.

Find out more about this feature here.

Search Test Steps

Searching for text within Test Steps of a manual Test case is now easier with the addition of a dedicated search field.

The browser search might not be a good solution because not all the steps are loaded at a given time due to performance reasons. Also, the steps might be collapsed.

Therefore, we have now included this search field that allows you to search within all Test Steps. This includes all Xray native fields (e.g. Action, Data, Expected Result) and also Test Step custom fields.

Learn more here.

Support for Robot Framework 4.0

[Robot Framework 4.0](#) is a new major release with a lot of big new features such as the SKIP status and native IF/ELSE support as well as enhancements to type conversion and Libdoc.

These changes have not been possible without breaking backward compatibility in some cases.

Xray now supports the new Robot Framework 4.0 report format.

Find out more about this feature [here](#).

[blocked URL](#)

And more...

Document Generator

- [XRAY-6659](#): Xray Test Repository - Make the "Doc Gen" option available in the "All" and "Orphans" views
- [XRAY-6623](#): When disabling the Document Generator, the option in Xray Settings should disappear too
- [XRAY-7022](#): Add the possibility of defining if tests from a sub-folder must be exported using Doc. Generator on Test Plan board
- [XRAY-7019](#): Add the possibility of defining if tests from a sub-folder must be exported using Doc. Generator on Test Repository board
- [XRAY-7193](#): As a Jira admin, I should be able to enable/disable Doc Gen at the project level
- [XRAY-7618](#): As a user, I should be able to use Doc Gen on Agile Boards
- [XRAY-7024](#): Provide a mapping to print the folder name which the test belongs to when the test is being exported from the Test Plan or Test Repository board

[blocked URL](#)

Just one more, yet <u>important</u>, thing...

Documentation is always important and sometimes you may miss it. We added a set of relevant articles to our already extensive documentation.

In this release, you may find new articles to facilitate integration with well-known CI/CD tools, including Travis CI and GitHub. We added also tutorials for Playwright, Cypress, WebdriverIO with corresponding open-source repositories, so you can easily try out these frameworks and integrate with Xray to have visibility of test automation results. You may find the code for these tutorials, and more, in our GitHub account. Check them out and let us know your feedback 🙂

A new section was created for Model-Based testing along with some tutorials.

Here's a summary of all the relevant articles that we've added and updated recently:

- **Tutorials with automation tools & frameworks**
    - Integration with Travis CI
    - Integration with Ranorex
    - Integration with GitHub
    - Testing web applications using Playwright
    - Testing using Cypress and Cucumber in JavaScript
    - Testing Node.js apps using Cucumber.js in JavaScript
    - Testing web applications using Mocha and WebdriverIO
    - Testing using WebDriverIO and Cucumber in JavaScript
    - TTT: Model-Based Testing
        - Model-Based Testing using GraphWalker and Java
        - Model-Based Testing using AltWalker and Python
    - TTT: API Testing
        - Testing REST services and APIs in Java using REST Assured and JUnit
- **Integration with other Jira apps**
    - Integration with Automation for Jira (added examples for implementing automation rules in different ecosystems and CI/CD tools, including Azure DevOps, Travis CI, etc)
- **Integrations from the community**
    - Integrations from the community and other products (included a GitHub Action to interact with Xray, from within GitHub)

blocked URL

All New Features and Bug Fixes in this release

Key	Summary	T	Created	P	Status	Resolution

No issues found