

Document Generator Template: Test Executions List Report

- [Purpose](#)
- [Output Example\(s\)](#)
- [How to use](#)
 - [Source data](#)
 - [Output format](#)
 - [Report assumptions](#)
 - [Usage examples](#)
 - [Export Test Executions of your project](#)
 - [Export Test Executions associated with a given fix version](#)
 - [Export Test Executions obtained on a given Test Environment](#)
 - [Export Test Execution from its detailed view](#)
- [Understanding the report](#)
 - [Layout](#)
 - ["Test Executions" sheet](#)
- [Customizing the report](#)
 - [Exercise 1: add a field from the related Test Execution issue](#)
 - [Exercise 2: add more details to the Test Plan column](#)
- [Performance](#)

Purpose

This report lists some details of the selected Test Executions in Xray, enabling them to be extracted in an Excel format. With this ability to extract the report you can use it for analysis of trends and current testing status, or process this information to generate some metrics, for example, or even share it with someone else who still needs access to Jira.

Possible usage scenarios:

- see all the Test Run statuses count for a given Test Execution
- see the linked Test Plans and Defects
- track the Test Executions that are taking the most time

Output Example(s)

The following table shows an example of the columns/rows you should expect.

Key	Summary	Fix Version	Revisio	Begin Date	End Date	Test Environm	Test Plan	Defects	Elapsed Time	#Test R	Passed	Passed (%)	Failed	Failed (%)	Executing	Executing (%)	To do	To do (%)	Aborted	Aborted (%)
BOOK-499	Ad-hoc execution for Manual Test	None	None			[]	BOOK-30		00:00:27	1	0	0	1	100	0	0	0	0	0	0
BOOK-500	Test Execution for Test Plan BOOK-30	1.0.2.0	123	18-02-2019 14:03:00	22-02-2019 14:03:00	[chrome, windows]	BOOK-30, BOOK-522	BOOK-510, BOOK-514	01:14:52	2	0	0	2	100	0	0	0	0	0	0
BOOK-505	Test Execution for Test Plan BOOK-30	1	None	18-02-2019 14:03:00	22-02-2019 14:03:00	[]	BOOK-30			1	0	0	0	0	1	100	0	0	0	0
BOOK-509	Ad-hoc execution for Cucumber Test As a user, None	None	None			[]			00:00:06	1	0	0	1	100	0	0	0	0	0	0
BOOK-513	Test Execution for Test Plan BOOK-30	1	None	18-02-2019 14:03:00	22-02-2019 14:03:00	[]	BOOK-30			5	1	20	0	0	0	0	3	60	1	20
BOOK-518	Ad-hoc execution for Validate addition feature: 2		None			[]				1	0	0	0	0	0	0	1	100	0	0
BOOK-519	Ad-hoc execution for Validate addition feature: 2		None			[]				0	0	0	0	0	0	0	0	0	0	0
BOOK-520	Addition validation	2	Revision Beta	21-11-2022 15:26:00	22-11-2022 15:26:00	[chrome, windows]	BOOK-30	BOOK-521	00:00:00	5	2	40	1	20	0	0	2	40	0	0
BOOK-523	Test Execution for Test Plan BOOK-522	2	Revision Beta	28-11-2022 15:37:00	05-12-2022 15:37:00	[chrome, windows]	BOOK-522	BOOK-524		4	2	50	1	25	1	25	0	0	0	0
BOOK-531	Test Execution for Test Plan BOOK-530	v4.0	None			[Chrome, windows]	BOOK-530			3	1	33.4	0	0	1	33.3	1	33.3	0	0

How to use

This report can be generated from different places/contexts, including:

- Issue view screen
- Issue search page (main search page or as a bulk operation)



Learn more

General information about all the existing places available to export from and how to perform it is available on the [Exporting](#) page.

Source data

This report is applicable to:

- 1 or more Test Execution issues

Output format

The standard output format is .XLSX so you can open it in Microsoft Excel, Google Sheets, and other tools compatible with this format. From those tools, you can generate a .CSV file.

Report assumptions

The template has a set of assumptions that you have to make sure your Jira/Xray environment complies with:

1. Issue types having the name: "Test Execution".

If any of these assumptions is not met, you need to update the template or the environment accordingly.

Usage examples

Export Test Executions of your project

1. from the Issue Navigator/Search, search by the issueType (i.e., "Test Execution") from your project (e.g., "BOOK") and then use bulk export or ExportDocument Generator

example of JQL expression to use

```
project = "BOOK" and issuetype="Test Execution" order BY created DESC
```

Export Test Executions associated with a given fix version

1. from the Issue Navigator/Search, search by the release (i.e., "fixVersion") of your project (e.g., "EWB") and then use bulk export or Export->Document Generator

example of JQL expression to use

```
project = "BOOK" and issuetype= "Test Execution" and fixVersion=1.2 order BY created DESC
```

Export Test Executions obtained on a given Test Environment

1. from the Issue Navigator/Search, search by Test Executions assigned to that Test Environment (e.g., "chrome") and then use bulk export or Export->Document Generator

example of JQL expression to use

```
project = "BOOK" and issuetype= "Test Execution" and testEnvironments = chrome order BY created DESC
```

Export Test Execution from its detailed view

1. open the Test Execution issue and export it using this template

Understanding the report

The report shows information about the Test Executions in a list form.

Layout

The report is composed of one sheet with the information on the "Test Executions".

By default, and to avoid overload/redundancy of information, only the "Test Executions" are rendered; all the other types will not be parsed.

"Test Executions" sheet

This sheet will present a line per each Test Execution.

Column	Notes
Key	Issue key of the Test Execution
Summary	Summary of the Test Execution
Fix versions	Fix version(s) defined in the Test Execution
Revision	Revision defined in the Test Execution
Begin Date	Timestamp of when the Test Execution started, in the format "dd-MM-yyyy hh:mm:ss"
End Date	Timestamp of when the Test Execution ended, in the format "dd-MM-yyyy hh:mm:ss"
Test Environment	Test Environment(s) defined in the Test Execution
Test Plan	Test Plan(s) linked to the Test Execution
Defects	Defects linked to the Test Execution (at either the test run or the test step level)
Elapsed Time	Sum of elapsed time of all associated test runs in HH:MM:SS format. Please note that it will output "00:00:00" when there are no tests associated with the execution (or your executions are really fast) and " " when there are tests in progress (i.e. "Executing" or "To Do" status).
#Test Runs	Number of test runs that are part of the Test Execution.
Passed	Number of runs in the passed status.
Passed (%)	Percentage of runs in the passed status.
Failed	Number of runs in the failed status.
Failed (%)	Percentage of runs in the failed status.
Executing	Number of runs in the executing status.
Executing (%)	Percentage of runs in the executing status.
To do	Number of runs in the to do status.
To do (%)	Percentage of runs in the to do status.
Aborted	Number of runs in the aborted status.
Aborted (%)	Percentage of runs in the aborted status.

Customizing the report

The common customization actions are:

- adding/removing columns
- changing the level of detail in the columns

Exercise 1: add a field from the related Test Execution issue

Let's say we have a "Description" field on the Test Execution issue that we want to show on the report.

We can copy the column "Summary" and adapt it.

1. Copy the "Summary" column
2. Insert the copied content in a column next to the "Fix Versions"
 - a. Change the header of the column to be "Description"
 - b. Change the cell content from `${Summary}` to `${Description}`
 - c. Save the template and upload it

As this report is column-based, if some columns are not relevant to you, you should be able to delete them. Make sure that no temporary variables are created in the cells of those columns that are used in other subsequent columns.

Exercise 2: add more details to the Test Plan column

Let's say you want to see not only the Test Plan Key, but also the Test Plan Summary, and make it a hyperlink while you are at it.

1. Navigate to the "Test Plan" column
2. Edit the statement from `${Test Plan}` to `${link:title=${Test Plan},href=${BaseURL}/browse/${Test Plan}}`

You can apply similar changes to the Defects column by editing the following statements:

- `${TestRuns[n].ExecutionDefects[d].Key}` (line 6)
- `${TestRuns[n].Iterations[it].TestSteps[r].Defects[dc].Key}` (line 22)

with the syntax like (the example is for line 6):

```
@{title=${TestRuns[n].ExecutionDefects[d].Key}|href=${BaseURL}/browse/${TestRuns[n].ExecutionDefects[d].Key}} | ${TestRuns[n].ExecutionDefects[d].Summary} | ${TestRuns[n].ExecutionDefects[d].Priority}
```

Keep in mind that only 1 hyperlink will be active per Excel cell (the one associated with the first line item). Also, `${Test Plan}` value is treated as a single list, so if you have multiple test plans linked to the same execution and you want to separate the urls, you will need to further customize the code from the example above. On a related note, as you probably noticed, Test Plan and Test Environment are displayed slightly differently from Defects by default - comma separation, square brackets (the DocGen version has the brackets for empty Test Plan cells, doesn't have them for non-empty Test Environments).

You can further finetune the content and formatting via JavaScript, you can find more useful snippets in [this tutorial for Document Generator](#).

Performance

Performance can be impacted by the information that is rendered and by how that information is collected/processed.

The number of Test Executions and Test Runs, depending on scenarios, can be considerably high, especially with CI/CD. As this report sums up quite a lot of information, please use it wisely.