Xporter Template: Test Runs List Report

Purpose

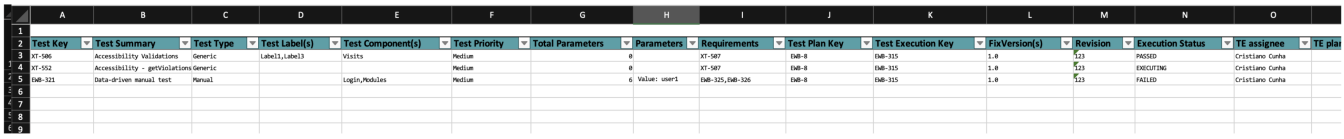
This report enables you to extract a list of Test Runs related to the input issues, so you can use it for analysis of trends, current testing status, or process this information to generate some metrics, for example, or even share it with someone else that hasn't access to Jira.

Possible usage scenarios:

- see all the Test Runs for a given Test Execution, Test Plan, Test, Test Set, or Story
- see the failed Test Runs and understand what happened and its impacts at high-level
- see the defects linked to failed Test Runs
- see the requirements linked to failed Test Runs
- see the failed Test Runs that have no defects
- track the Test Runs that are taking most time

Output Example(s)

The following table shows an example of the columns/rows you should expect, in this for the "Test Runs" sheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1																
2	Test Key	Test Summary	Test Type	Test Label(s)	Test Component(s)	Test Priority	Total Parameters	Parameters	Requirements	Test Plan Key	Test Execution Key	FixVersion(s)	Revision	Execution Status	TE assignee	TE pla
3	XT-506	Accessibility Validations	Generic	Label1,Label3	Visits	Medium	0		XT-507	EWB-8	EWB-315	1.0	123	PASSED	Cristiano Cunha	
4	XT-552	Accessibility - getViolations	Generic			Medium	0		XT-507	EWB-8	EWB-315	1.0	123	EXECUTING	Cristiano Cunha	
5	EWB-321	Data-driven manual test	Manual		Login,Modules	Medium	6	Value: user1	EWB-325,EWB-326	EWB-8	EWB-315	1.0	123	FAILED	Cristiano Cunha	
6																
7																
8																
9																

In more depth...

Test Key	Test Summary	Test Type	Test Label (s)	Test Component (s)	Test Priority	Total Parameters	Parameters	Requirements	Test Plan Key	Test Execution Key	FixVersion (s)	Revision	Execu Statu
XT-506	Accessibility Validations	Generic	Label1, Label3	Visits	Medium	0		XT-507	EWB-8	EWB-315	1.0	123	PASSEI
XT-552	Accessibility - getViolations	Generic			Medium	0		XT-507	EWB-8	EWB-315	1.0	123	EXECU

EWB-321	Data-driven manual test	Manual		Login,Modules	Medium	6	Key: username Value: user1 _____ Key: password Value: pass1 _____ Key: username Value: user2 _____ Key: password Value: pass2 _____ Key: username Value: user3 _____ Key: password Value: pass3 _____	EWB-325,EWB-326	EWB-8	EWB-315	1.0	123	FAILED

How to use

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

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

Learn more

General information about all the existing places available to export from, and how to perform it, is available in the Exporting page.

Source data

This report is applicable to:

- 1 or more Test Plan issues
- 1 or more Test Set issues
- 1 or more Test Execution issues
- 1 or more Test issues
- 1 or more Story issues
- a combination of the previous

Please note: to avoid redundancy in the output, we recommend not mixing source types, especially if the entities are also linked with each other (e.g. passing the issue keys for Stories and Test Executions linked to them to the template at the same time)

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 may be able to generate a .CSV file.

Report assumptions

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

1. Issue types having these names
    a. "Test", "Test Set", "Test Plan", "Test Execution", "Story"

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

Usage examples

Export Test Runs associated with a given release

1. from the Issue Navigator/Search, search by the release (i.e., "fixVersion") of your project (e.g., "BOOK") and then use bulk export

example of JQL expression to use
`project = BOOK and fixVersion = 1.0 and issuetype = "Test Execution"`

Export Test Runs obtained in the context of a given Test Plan

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

Export Test Runs 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

example of JQL expression to use
`project = BOOK and issuetype = "Test Execution" and  "Test Environments" = chrome`

Understanding the report

The report shows information about the Test Runs for the input issues provided.

Layout

The report is composed by several sheets. The information on the "Test Runs" and "Test Runs with iterations" is mostly the same; please see details ahead.

By default, and to avoid overload/redundancy of information, only the "Test Runs" and the "Input Issues" sheets will be rendered; you can change this behavior on the template (more info ahead).

"Test Runs" sheet

This sheet will present a line per each Test Run, no matter the type of Test Run.

Column	Notes
Test Key	issue key of the Test associated to this Test Run
Test Summary	Summary of the Test associated to this Test Run
Test Type	test type of the Test associated to this Test Run (e.g., Manual, Cucumber, Generic)
Test Label(s)	labels of the Test associated to the Test Run, delimited by comma (e.g., "UI,selenium")
Test Component (s)	Components field of the Test associated to the Test Run, delimited by comma (e.g., "core,backend")
Test Priority	Priority field of the Test associated to the Test Run, delimited by comma (e.g., "Major")
Total Parameters	number of test parameters, for parameterized tests
Parameters	list of test parameters, for parameterized tests (e.g., key: <param_name> value: <param_value>")
Requirements	list of covered items (e.g., requirements, stories) by the Test associated to the Test Run, delimited by comma (e.g., "CALC-1")
Test Plan Key	issue key of the Test Plan(s), associated to the Test Execution, that in turn is associated to the Test Run
Test Execution Key	issue key of the Test Execution associated to the Test Run
FixVersion(s)	issue key of the Test Execution associated to the Test Run

Revision	Revision field of the Test Execution associated to the Test Run
Execution Status	the status/result reported for this Test Run
TE assignee	the display name of the assignee of the Test Execution associated to the Test Run
TE planned begin date	planned date for starting the Test Execution, in the format "dd/MM/yyyy"
TE planned end date	planned date for finishing the Test Execution, in the format "dd/MM/yyyy"
Assignee	(full) name of the user assigned to the Test Run
Executed by	(full) name of the user who has executed the Test Run
Test Environment(s)	Test Environment(s) of the Test Execution associated to the Test Run, delimited by comma (e.g., "firefox", "edge,windows")
Started on	timestamp of when the Test Run started, in the format "dd/MM/yyyy hh:mm:ss"
Finished on	timestamp of when the Test Run finished, in the format "dd/MM/yyyy hh:mm:ss"
Elapsed	elapsed time, in the format "hh:mm:ss", for the Test Run; for data-driven tests, this corresponds to the overall Test Run elapsed time
Elapsed (sec)	elapsed time, in seconds, for the Test Run; for data-driven tests, this corresponds to the overall Test Run elapsed time
Comment	Test Run comment
Total Defects	number of unique defects linked/reported to the Test Run, either globally, at step level, including from iterations in data-driven tests
Defects	list of issue keys of unique defects linked/reported to the Test Run, either globally, at step level, including from iterations in data-driven tests, delimited by comma

"Test Runs including iterations" sheet

This sheet will present:

- a line per each Test Run that is non data-driven (i.e., a standard run)
- for data-driven tests, a line per each iteration (i.e., data row) of the Test Run

Therefore, this sheet is like a superset of the "Test Runs" sheet, where data-driven test runs are expanded to multiple rows.

The fields displayed in this sheet are similar to the previous one.

"Input Issues" sheet

This sheet list the issues that were used as input for generating the report.

Column	Notes
Key	input issue's key
Issue Type	name of the issue type of the input issue (e.g., "Test", "Test Execution")
Summary	Summary field of the input issue
Priority	Priority field of the input issue
Component(s)	Components field of the input issue, delimited by comma (e.g., "core,backend")
Priority	Priority field of the input issue (e.g., "Major")

Status	Workflows status the input issue (e.g., "Reviewed")
Assignee	the display name of the assignee of the input issue

Customizing the report

Sections that can be hidden or shown

The report has some variables/flags that can be used to show or hide some sections whose logic is already implemented in the template.

These variables are defined at the top of each sheet, at the report template; the variables are scoped just to the current sheet.

On the template, use one of these values for flag type of variables:

- **0**: to hide a section
- **1**: to show a section

The format for other types of variables is detailed ahead.

Example of setting a variable to, in this case, render information on the sheet "Test Runs including iterations"

${set(renderThisSheet, 1)}

Variable/flag	Purpose	default	example(s)
renderThisSheet	render this sheet • format: 0 or 1	0 (shown)	${set(renderThisSheet, 1)}
statusesToInclude	render Test Runs whose reported status is one in this list (delimited by comma); use an empty string '' to include all statuses • format: '<status1>,<status2>"	'' (i.e., all statuses)	${set(statusesToInclude, 'FAIL')} ${set(statusesToInclude, 'FAIL, EXECUTING')} ${set(statusesToInclude, '')}

Adding or removing information to/from the report

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

To add additional information, usually we're thinking of adding:

- fields of the Test Run itself
- fields of the Test associated with the Test Run
- fields of the Test Execution associated with the Test Run

Eventually, also:

- fields of the Test Plan(s) associated with the Test Execution that is associated with the Test Run
- fields of the covered issue(s) associated with the Test that is associated with the Test Run

The later may be harder to implement, so we won't consider them here.

Exercise: add a field from the related Test issue

Let's say we have a "Severity" field on the Test issue that is connected to each Test Run, and that we want to show it on the report.

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

1. insert column
2. on the "Test Runs" sheet,
    a. copy "Test Summary" (i.e., insert a column next to it and copy the values from the existing "Test Summary" column)
    b. change
        i. ${TestRuns[n].Summary} to ${TestRuns[n].Severity}

        ii. ${TestExecutions[j].TestRuns[n].Severity} to ${TestExecutions[j].TestRuns[n].Severity}
3. follow a similar approach for the "Test Runs including iterations" sheet

Exercise: add total Evidence count

Let's say we want to know the total count of evidence items for each run and iteration. You can add the column next to the "Comment" one, then the code logic is similar to the defect count. Please see the snippet below (this exact syntax is for the "Test Runs" tab):

EvidenceSnippet

```
${set(totalStepEvidenceCount, 0)}
${set(totalEvidenceCount, 0)}
#{if (%{!${TestExecutions[j].TestRuns[n].IsDataDriven}})}
#{for m=TestExecutions[j].TestRuns[n].TestStepsCount}
#{for l=TestExecutions[j].TestRuns[n].TestSteps[m].EvidencesCount}
${set(totalStepEvidenceCount,%{${totalStepEvidenceCount} + 1 })}
#{end}
#{end}
${set(totalEvidenceCount,%{${totalStepEvidenceCount} + ${TestExecutions[j].TestRuns[n].ExecutionEvidencesCount}
})}
#{end}
#{if (%{${TestExecutions[j].TestRuns[n].IsDataDriven}})}
#{for m=TestExecutions[j].TestRuns[n].IterationsCount}
#{for k=TestExecutions[j].TestRuns[n].Iterations[m].TestStepsCount}
#{for l=TestExecutions[j].TestRuns[n].Iterations[m].TestSteps[k].EvidencesCount}
${set(totalStepEvidenceCount,%{${totalStepEvidenceCount} + 1 })}
#{end}
#{end}
${set(totalEvidenceCount,%{${totalStepEvidenceCount} + ${TestExecutions[j].TestRuns[n].ExecutionEvidencesCount}
})}
#{end}
#{end}
${totalEvidenceCount}
```

Keep in mind that the core part of the path to the target (i.e. "TestExecutions[j].TestRuns[n]." above) is dependent on the source issue type, so you will need to adjust it based on the row in the template. You can refer to other cells in the same row for the path syntax.

For the "Test Runs including Iterations" tab, you will only need the content of one of the two "if" loops for each row and, for the data-driven rows, you will need to drop the additional "for" loop (i.e. "#{for m=TestExecutions[j].TestRuns[n].IterationsCount}" from the second "if" loop in the snippet above).

Performance

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

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

Data-driven tests may also add an overhead, as iterations need to be individually processed, for collecting all the reported/linked defects for example.

Some tips

- Use the variables/flags to adjust sections or the Test Runs that will be processed/shown in the report; more info in "Customizing the report"
- limit the number of input issues; in Xporter there's a global setting for this purpose

Known limitations

- Test Runs comments are not formatted
- Gherkin Scenario Outlines are not considered as data-driven (i.e., only one Test Run will appear)