

Importing and Exporting Tests (CSV)



Do you want to migrate tests from other tools? If you are able to export them to a Excel/CSV file, then you may import the tests to Jira using also using the instructions below.

If you need guidance on a specific use case, please [contact us](#).

- [Import Tests](#)
 - [Importing Manual Tests](#)
 - [Importing Cucumber Tests](#)
 - [Importing Automated Generic Tests](#)
- [Export Tests](#)
- [Import Pre-Conditions, Test Sets, Test Executions and Test Plans](#)
 - [Pre-Conditions](#)
 - [Test Sets](#)
 - [Test Executions](#)
 - [Test Plans](#)
- [Copying Tests from one Jira server to another one](#)
- [Importing execution-related information from a legacy \(non-Jira\) test management tool to Xray on Jira](#)

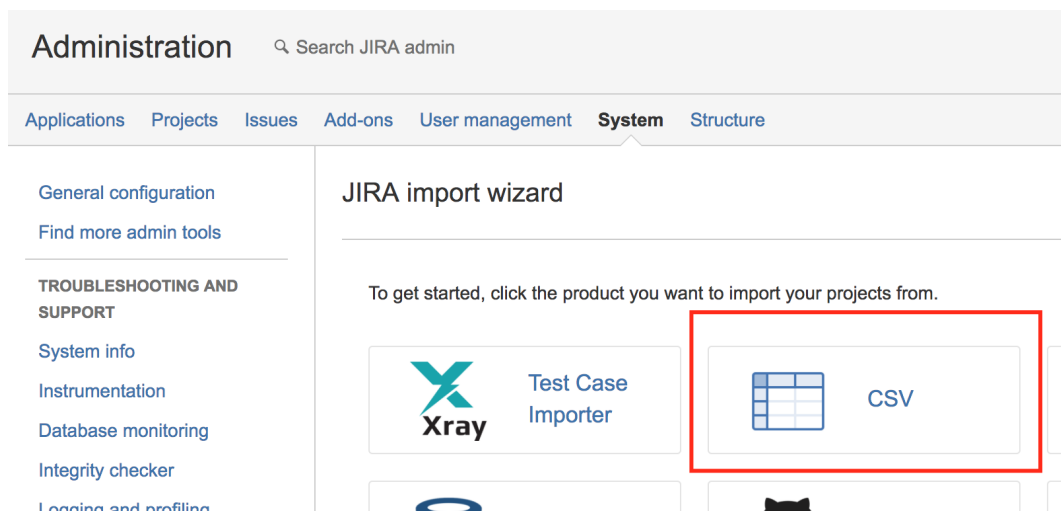
Import Tests

You can import Test definitions from external sources using CSV files, either by using [Xray's Test Case importer](#) (**recommended**) or Jira's native CSV importer plugin (see [Importing Data from CSV](#)).

This page explaining the later procedure, which is more generic and supports importing of all Test types. **However**, for manual Tests you should use [Xray's Test Case importer](#).

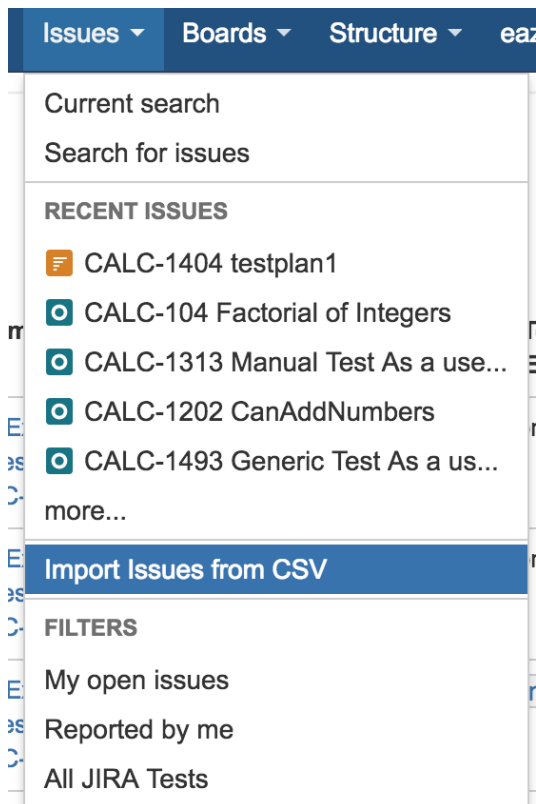
To import issues using Jira's CSV importer, as an administrator, you have to:

1. Log in to Jira as a user with the **Jira Administrators** [global permission](#).
2. Select **Administration > System > Import & Export > External System Import > Import**, and then specify the input CSV file.



As a standard user, you can also import issues:

1. Go to the top **Issues** menu.
2. Select **Import Issues from CSV**.



Aside from the native issue columns like *Summary* and *Description*, you need to specify in the CSV file additional columns related to Xray Test issues.

- **Issue Type** - must be present in your CSV and must contain the name of Xray Test issue type, which is, by default, "**Test**". Alternately, this column can contain the Test issue type ID.
- **Test Type** - required for each Xray Test Issue. This field can have the following values: Manual, Automated[Cucumber] or Automated[Generic]

There are some columns that you also need to specify, depending on the type of Test as described in the next sections.



Links to requirements

When importing Tests, you can specify the links to the requirements that each Test validates. For that purpose, the columns must be mapped as **Link "Tests"**.

If the Test covers multiple requirements, then multiple CSV columns must be used, each one being mapped in the same way.

Link to requirement
(e.g. CALC-881)



Link "Tests"

Importing Manual Tests



Recommended

If you need to import Manual Tests, you should use [Xray's Test Case importer](#) instead since it supports a more user-friendly CSV file layout for describing Test steps.



Please be aware that it may be necessary to update **Jira Importers Plugin (JIM)** in order to map all custom fields.

If you are importing Tests from a CSV and you are linking your tests to an already existing requirement, you have to **Clear the Requirement Status Cache**. Please check [Importing and Exporting Tests \(CSV\)](#) for more information.

In order to import Manual Tests in a CSV file to Jira, you need define the following columns:

- **"Test Type"** is "Manual"
- **"Manual Test Steps"** is the custom field in JSON list format representing all your manual steps. Here is an example for the **"Manual Test Steps"** custom field:

```
[
  {
    "index": 0,
    "step": "Step 1",
    "data": "input Data 1",
    "result": "Excepted result 1"
  },
  {
    "index": 1,
    "step": "Step 2",
    "data": "input Data 2",
    "result": "Excepted result 2"
  },
  {
    "index": 2,
    "step": "Step 3",
    "data": "input Data 3",
    "result": "Excepted result 3"
  },
  {
    "index": 3,
    "step": "Step 4",
    "data": "input Data 4",
    "result": "Excepted result 4"
  }
]
```

Example of CSV file with manual tests

```
Summary; Assignee; Reporter; Issue Type; Description; Test Type; Manual Test Steps
"Test issue 1"; admin; admin; 10000; "This is a Test issue"; "Manual"; "[ { \"index\": 0, \"step\": \"Step 1\",
\"data\": \"input Data 1\", \"result\": \"Excepted result 1\" }, { \"index\": 1, \"step\": \"Step 2\",
\"data\": \"input Data 2\", \"result\": \"Excepted result 2\" } ]"
"Test issue 2"; admin; admin; 10000; "This is a Test issue"; "Manual"; "[ { \"index\": 0, \"step\": \"Step 1\",
\"data\": \"input Data 1\", \"result\": \"Excepted result 1\" } ]"
```



How to optimize this process

Some organizations already have Manual Test definitions in spreadsheets. These spreadsheets normally define all Tests to execute and within each Test, their steps. This is the main reason why [Xray's Test Case importer](#) is a better alternative for importing Manual Tests.

Before Xray for Jira 2.1, you could use a [special converter tool](#) (now **deprecated**) for converting these spreadsheet Test definitions, more specifically the Test Steps, into the JSON format above expected by Xray and Jira.

Importing Cucumber Tests

For Cucumber Test issues, you need to define the following columns in your CSV:

- **"Test Type"** is "Automated[Cucumber]"
- **"Cucumber Test Type"** is either "Scenario" or "Scenario Outline"
- **"Cucumber Scenario"** is the contents of a Cucumber Scenario.

Importing Automated Generic Tests

For automated generic Test issues, you need to define the following columns in your CSV:

- **"Test Type"** is "Automated[Generic]"
- **"Generic Test Definition"** is the test definition.

Export Tests

Jira lets you export your data in CSV or Excel (file is saved as *XLS*, but you can open it with your spreadsheet tool and export it to CSV). Xray custom fields are compatible with this function. Even Manual Test Steps field can be exported in a JSON format so that it can be imported back to Jira directly.



Please note

Since Jira 7.2, Jira **only** provided the ability to export issues to CSV in a limited way; recent versions do not support exporting to Excel.

If you need to export issues, including your Tests, to Excel or fully customizable templates, then we recommend you use the [Xporter app](#).

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due	Manual Test Steps	Test Execution Status	Test Set Status
	XTP-6	As a user I can login	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":4,"index":1,"step":"","data":"","result":"I can see the app login page appearing","attachments":[]}]]		
	XTP-5	As a admin I can create a user	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":5,"index":2,"step":"","data":"","result":"I can see the menu screen","attachments":[]}]]		
	XTP-4	Test app Menu	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":1,"index":1,"step":"","data":"","result":"The app is now on top of all other app","attachments":[]}]]		
	XTP-3	Test Execution for create user	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":3,"index":3,"step":"","data":"","result":"I can see the list of users with the newly created user on it","attachments":[]}]]		
	XTP-2	Test - As a user I can login on app	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":3,"index":3,"step":"","data":"","result":"I can see the list of users with the newly created user on it","attachments":[]}]]		
	XTP-1	Test - I can create a new user with the app	admin	admin		open	Unresolved	17/Sep/15	17/Sep/15		[[{"id":1,"index":1,"step":"","data":"","result":"The app is now on top of all other app","attachments":[]}]]		

Fields

Select the CSV fields to import, then set how you would like these converted to fields in JIRA. You can optionally map field values on the next screen.

CSV Field		JIRA field	Map field value
Assignee (e.g. admin)	→	Assignee	<input type="checkbox"/>
Condition (e.g. calculator must be turned on)	→	Conditions	<input type="checkbox"/>
Description (e.g. This is a PreCondition issue)	→	Description	<input type="checkbox"/>
Issue Type (e.g. 10)	→	Issue Type	<input type="checkbox"/>
Reporter (e.g. admin)	→	Reporter	<input type="checkbox"/>
Summary (e.g. Standalone PreCondition)	→	Summary	<input type="checkbox"/>
Tests associated with Pre Condition (e.g. First row doesn't have a value)	→	Tests association with a Pre-Con...	<input type="checkbox"/>
Type (e.g. Manual)	→	Pre-Condition Type	<input type="checkbox"/>

Test Sets

You can specify the Tests by using the **Tests associated with Test Set** mapping upon importing. Use the comma (",") delimiter to specify multiple tests.

Example of CSV file with manual tests

```
Summary; Assignee; Reporter; Issue Type; Description; Tests associated with Test Set
"Test Set without Tests"; admin; admin; 8; "This is a Test Set issue";
"Test Set with 2 Tests"; admin; admin; 8; "This is a Test Set issue"; CALC-908,CALC-887
```

Fields

Select the CSV fields to import, then set how you would like these converted to fields in JIRA. You can optionally map field values on the next screen.

CSV Field		JIRA field	Map field value
Assignee (e.g. admin)	→	Assignee	<input type="checkbox"/>
Description (e.g. This is a Test Set issue)	→	Description	<input type="checkbox"/>
Issue Type (e.g. 9)	→	Issue Type	<input type="checkbox"/>
Reporter (e.g. admin)	→	Reporter	<input type="checkbox"/>
Summary (e.g. Test Set without Tests)	→	Summary	<input type="checkbox"/>
Tests associated with Test Set (e.g. First row doesn't have a value)	→	Tests association with a Test Set	<input type="checkbox"/>

Test Executions

You can specify the Tests by using the **Tests association with Test Execution** mapping upon importing. Use the comma (",") delimiter to specify multiple tests.

Example of CSV file with manual tests

```
Fix Version; Summary; Assignee; Reporter; Issue Type; Description; Test Environments; Tests associated with
Test Execution
"3.0"; "Test Execution without Tests"; admin; admin; 9; "This is a Test Execution issue";
"3.0"; "Test Execution with 2 Tests"; admin; admin; 9; "This is a Test Execution issue"; Android; CALC-908,CALC-
887
```

Fields

Select the CSV fields to import, then set how you would like these converted to fields in JIRA. You can optionally map field values on the next screen.

CSV Field		JIRA field	Map field value
Assignee (e.g. admin)	→	Assignee	<input type="checkbox"/>
Description (e.g. This is a Test Execution issue)	→	Description	<input type="checkbox"/>
Fix Version (e.g. 3.0)	→	Fix Version/s	<input type="checkbox"/>
Issue Type (e.g. 9)	→	Issue Type	<input type="checkbox"/>
Reporter (e.g. admin)	→	Reporter	<input type="checkbox"/>
Summary (e.g. Test Execution without Tests)	→	Summary	<input type="checkbox"/>
Test Environments (e.g. First row doesn't have a value)	→	Test Environments	<input type="checkbox"/>
Tests associated with Test Execution (e.g. First row doesn't have a value)	→	Tests association with a Test Exe..	<input type="checkbox"/>

Test Plans

You can specify the Tests by using the **Tests associated with Test Plan** mapping upon importing. Use the comma (",") delimiter to specify multiple tests.

Example

```
Fix Version; Summary; Assignee; Reporter; Issue Type; Description; Tests associated with Test Plan
"3.0"; "Test Plan without Tests"; admin; admin; 10; "This is a Test Plan issue";
"3.0"; "Test Plan with 2 Tests"; admin; admin; 10; "This is a Test Plan issue"; CALC-908,CALC-887
```

Copying Tests from one Jira server to another one

Sometimes, you may want to copy Tests specified in some other Jira server, such as a staging or testing environment.

If you need to "migrate" (i.e., copy) the Tests to another Jira server, you may follow the instructions above for exporting Tests to CSV and to import them from CSV.

The general procedure would be:

1. In the source Jira server
 - a. filter out the the relevant Test issues in the Issues search screen
 - b. make visible the columns you want to migrate (don't forget to include the Summary, Issue Type, Test Type, Manual Test Steps and other mandatory fields)
 - c. export visible fields to CSV
2. In the destination JIRA server
 - a. import Tests from CSV using Jira's built-in CSV importer (don't forget to map all the mandatory fields)

What you will be able to copy:

- Test-related fields (e.g., summary, labels, etc.)
- Test Type (mandatory)

- Manual Test Steps (mandatory for Manual tests)

What you won't be able to copy:

- attachments (global- or step-level attachments)
- related entities and links (because they won't exist in the destination Jira server)

Please note

This procedure allows you to migrate the essential Test specifications. If you need to migrate test runs and other information, then this procedure is not applicable. You may use the REST API, but it will require some work from your side since there is no direct way to migrate it.

Importing execution-related information from a legacy (non-Jira) test management tool to Xray on Jira

Although Xray allows you to import execution-related information, Xray does not currently provide an out-of-the-box solution for migrating execution-related information along with the corresponding tests from other systems.

Please note

This approach requires some development effort from your side.

1. Migrate Tests specification

- From CSV using Test Case Importer; supports attachment, but you will miss the mapping information between original_idócreated_issue_key; you will have to somehow manage it.
- Programmatically, create the Test issues using Jira's REST API (some examples are in <http://confluence.xpand-addons.com/display/XRAY/Tests++REST#Tests-REST-CreatingandEditingTests-JIRARESTAPI>); this will allow you to easily identify the mapping original_idócreated_issue_key, which can be used afterwards.

2. Migrate organization (optional, not really necessary)

This depends on the legacy system organization. If it somehow groups tests, then use the Jira REST API to create the Test Set issues. After that, use Xray's REST API to add the Tests. You must know the Tests issue keys in advance.

3. Import results

- Create Test Executions with their respective Tests

Use the Jira REST API can be used to create the Test Execution issue. After that, use Xray's REST API (i.e., POST `/rest/raven/1.0/api/testexec/{testExecKey}/test`) to add the Tests. You must know the Tests issue keys in advance.

- Import results for each Test Execution

- Import just the overall test result, for each test run that is part of the test execution.
- Import detailed test run results, including per step information

Each approach can be achieved using the REST API, and the [POST /rest/raven/1.0/import/execution](#) endpoint. You just need to know in advance the Test Execution key and also the issues keys for each Test that is part of the Test Execution.