

# Importing Tests with (CSV)



Do you want to migrate tests from other tools? If you are able to export them to an Excel/CSV file, then you may import the tests to Jira using also using the instructions below. Note that [Xray's Test Case importer](#) is the **recommended tool** for this purpose and not Jira's native CSV importer as detailed here.

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

- [Import Tests](#)
  - [Importing Manual Tests](#)
  - [Importing Cucumber Tests](#)
  - [Importing Automated Generic Tests](#)
- [Import Pre-Conditions, Test Sets, Test Executions and Test Plans](#)
  - [Pre-Conditions](#)
  - [Test Sets](#)
  - [Test Executions](#)
  - [Sub 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](#)
- [Configuring maximum allowed attachment size](#)

## 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.

The screenshot shows the Jira Administration interface. The top navigation bar includes 'Jira', 'Dashboards', 'Projects', 'Issues', 'Boards', 'DbConsole', 'Tests', and a 'Create' button. The 'Administration' section is active, with a search bar and tabs for 'Applications', 'Projects', 'Issues', 'Manage apps', 'User management', 'Latest upgrade report', and 'System'. The left sidebar lists various configuration and support options. The main content area, titled 'Jira import wizard', instructs the user to click a product to import projects from. It displays six options: 'Test Case Importer' (XRAY), 'HP QC/ALM Importer' (XRAY), 'Zephyr for Jira Importer' (XRAY), 'CSV', 'JSON', and 'None of these'. Below these options, a note states: 'You are using version 9.1.2. For help with this plugin please contact [JIRA support](#). All registered trademarks or trademarks are property of their respective owners. Nothing on this page means that any third parties endorse the use of Atlassian products.'

As a standard user, you can also import issues:


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


Issues ▾Boards ▾Structure ▾eaz


Current search


Search for issues


RECENT ISSUES

 CALC-1404 testplan1

 CALC-104 Factorial of Integers

 CALC-1313 Manual Test As a use...

 CALC-1202 CanAddNumbers

 CALC-1493 Generic Test As a us...

more...

Import Issues from CSV

FILTERS

My open issues

Reported by me

All JIRA Tests

## Mapping Test Steps Custom fields

Please note that when using Manual Test Steps Custom field of Type "Date Picker" or "DateTime Picker", Xray uses the format set in the "Look and Feel" of your Jira instance.

### Date/Time Formats

Documentation on date/time formats can be found [online](#).

Time Format	<b>h:mm a</b> E.g. 4:55 AM	
Day Format	<b>EEEE h:mm a</b> E.g. Wednesday 4:55 AM	
Complete Date/Time Format	<b>dd MMM yyyy H:mm</b> E.g. 23 May 2007 4:55	<button>Revert</button>
Day/Month/Year Format	<b>dd MMM yyyy</b> E.g. 23 May 2007	<button>Revert</button>
Use ISO8601 standard in Date Picker	<b>No</b> <small>Turning it on will cause Monday to be the first day of week in the Date Picker, as specified by the ISO8601 standard</small>	

The format provided in the CSV configuration only refers to Jira custom fields and will not be used by Xray.

The Manual Test Step Custom fields must be configured and enabled in the target project or else Xray won't be able to map the fields and won't import any manual test step custom field



Please be aware that if one of the Test Step Custom fields is defined as mandatory in the target project and that same Custom Field is empty (without any value) in the CSV, after the Test Case importer process, the Test will be created successfully but without any of the steps.

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 Tests with \(CSV\)](#) for more information.

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

- **"Test Type"**, ensuring its value is "Manual"
- **"Manual Test Steps"**; this is the custom field in the JSON list format representing all your manual steps. Next follows an example of the **"Manual Test Steps"** custom field; due to CSV format, double quotes within this field have to be escaped whenever writing it into the CSV.

```
[
  {
    "index":1,
    "fields":{
      "Action":"Step 1 action",
      "Data":"",
      "Expected Result":"Expected result",
      "Limit date":"06 Jun 2020"
    },
    "attachments":[
      {
        "fileName":"Xray Unit Testing.docx",
        "fileIcon":"word.gif",
        "mimeType":"application/vnd.openxmlformats-officedocument.wordprocessingml.document",
        "fileIconAlt":"Microsoft Word",
        "fileSize":"506 kB",
        "numericalFileSize":517880,
        "created":"2020-06-04T22:46:21+01:00",
        "createdDate":"Jun 4, 2020 10:46:21 PM",
        "author":"admin",
        "authorFullName":"john",
        "filePath":"valid/local/path/to/server/inside/jira/directory"
      }
    ]
  },
  {
    "index":2,
    "fields":{
      "Action":"Step 2 action",
      "Data":"",
      "Expected Result":"Expected result",
      "Limit date":"07 Jun 2020"
    },
    "attachments":[
    ]
  }
]
```

This is a very basic example containing two test cases, having 2 steps each.

#### Example of CSV file with manual tests

```
Summary; Assignee; Reporter; Issue Type; Description; Test Type; Manual Test Steps
"Basic Test issue"; admin; admin; Test; "This is a basic Test issue"; "Manual"; "[{"index":1,"fields":{"action":"Step 1 action","data":"","expected result":"Expected result"},"attachments":[]}, {"index":2,"fields":{"action":"Step 2 action","data":"","expected result":"Expected result"},"attachments":[]}]"
```

This second example shows more complex/complete scenarios, detailing how to specify attachments, additional Xray step custom fields.

### 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":1,"fields":{"Action":"Step 1 action","Data":"","Expected Result":"Expected result"},"attachments":[{"fileName":"Xray Unit Testing.docx","fileIcon":"","word.gif","mimeType":"application/vnd.openxmlformats-officedocument.wordprocessingml.document","fileIconAlt":"Microsoft Word","fileSize":"506 kB","numericalFileSize":517880,"created":"2020-06-04T22:46:21+01:00","createdDate":"Jun 4, 2020 10:46:21 PM","author":"","authorFullName":"bruce wayne","filePath":"/jira/postgres/870/home/data/attachments/IMCSV/10000/IMCSV-1/xray_3883"}]}],{"index":2,"fields":{"Action":"Step 2 action","Data":"","Expected Result":"Expected result","Limit date":"07 Jun 2020"},"attachments":[]}]
"Test issue 2"; admin; admin; 10000; "This is a Test issue"; "Manual"; [{"index":1,"fields":{"Action":"Step 1 action","Data":"","Expected Result":"Expected result","Limit date":"06 Jun 2020"},"attachments":[]}, {"index":2,"fields":{"Action":"Step 2 action","Data":"","Expected Result":"Expected result","Limit date":"07 Jun 2020"},"attachments":[]}]
```



#### 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.

## Import Pre-Conditions, Test Sets, Test Executions and Test Plans

Since Xray uses Jira issue types for its core entities, it's also possible to import other entities such as Pre-Conditions, Test Sets, Sub Test Executions, Test Executions and Test Plans.

In the following examples, the Issue Type can either be specified as a string with the name of the Issue Type or with the corresponding *id* that can be found within "Issue Types" JIRA administration section.

### Pre-Conditions

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

### Example of CSV file with manual tests

```
Summary; Assignee; Reporter; Issue Type; Type; Condition; Description; Tests associated with Pre Condition
"Standalone PreCondition"; admin; admin; 10; Manual; "calculator must be turned on"; "This is a PreCondition issue";
"PreCondition linked to 2 Tests"; admin; admin; 10; Manual; "calculator must be scientific"; "This is a PreCondition issue"; CALC-779,CALC-756
"PreCondition linked to 1 Test"; admin; admin; 10; Manual; "calculator must be normal"; "This is a PreCondition issue"; CALC-779
```

### 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"/>

## Sub Test Executions

Sub Test Executions are sub-tasks of "requirements" (e.g. Story, Epic, other). Thus, the parent issue key must be provided; Jira's CSV also requires that an empty column mapped to the "Issue Id" to be passed in this case.

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 Sub Test Executions

```
Issue Id; Fix Version; Summary; Assignee; Reporter; Issue Type ; Parent issue (requirement); Description; Test
Environments; Tests associated with Test Execution
; "3.0"; "Sub Test Execution with 1 Test"; admin; admin; 10101; CALC-2541; "This is a very simple Sub Test
Execution issue"; Android; CALC-908
; "3.0"; "Sub Test Execution with 2 Tests"; admin; admin; Sub Test Execution; CALC-2541; "This is a Sub Test
Execution issue"; Android; CALC-908,CALC-2542
```

CSV Field		JIRA field	Map field value
Assignee (e.g. admin)	→	Assignee	<input type="checkbox"/>
Description (e.g. This is a Sub Test Execution issue)	→	Description	<input type="checkbox"/>
Fix Version (e.g. 3.0)	→	Fix Version/s	<input type="checkbox"/>
Issue Id (e.g. First row doesn't have a value)	→	Issue Id	<input type="checkbox"/>
Issue Type (e.g. 10101)	→	Issue Type	<input type="checkbox"/>
Parent issue (requirement) (e.g. CALC-2541)	→	Parent Id	<input type="checkbox"/>
Reporter (e.g. admin)	→	Reporter	<input type="checkbox"/>
Summary (e.g. Sub Test Execution with 2 Tests)	→	Summary	<input type="checkbox"/>
Test Environments (e.g. Android)	→	Test Environments	<input type="checkbox"/>
Tests associated with Test Execution (e.g. CALC-908,CALC-2542)	→	Tests association with a Test Ex...	<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 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)
- attachments (step-level) if present in the target Jira server directories and mapped in the Manual Test Step Json

What you won't be able to copy:

- 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 attachments, but you will miss the mapping information between original\_id and the created\_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\_id=>created\_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. On the other hand, in case you're using folders as means to organize Tests both in the original system and also in Xray, you may simply specify the folder as being mapped to the "Test Repository Path" Xray's custom field during the Test Case Importer's import process.

### 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.

## Configuring maximum allowed attachment size

The maximum allowed size for the Test Case Importer is the same as [Jira's](#) for CSV imports.

