

# Examples using Test Case Importer



## What you'll learn

### Prerequisites

- [Examples](#)
  - Different usages of Test Case Importer
  - [Import different test cases](#) (with different fields and options)
  - [Import manual test cases](#) (with different fields and options)
  - [Validate manually that the test cases are available](#)
  - [Import manual test cases with labels](#)
  - [Import manual test cases with Test Repository](#)
  - [Import manual test cases with Test Repository and link to requirement](#)
  - [Import manual test cases with custom Test Step and Components](#)
  - [Import manual test cases with attachments](#)

### Source code for this tutorial

- [Mapping priorities](#)
- [All examples are available in GitHub](#)
- [Update two manual test cases](#)
- [Tips](#)

## Overview

Test Case Importer is used to import test cases in a CSV format into Xray, this functionality is used to migrate from other tools or to import manual test cases written in other formats into Xray.

In this tutorial we will provide different examples of how to import test case in the CSV format into Xray.

Notice that until version 6.2 Test Case Importer only allow to import/update manual tests cases.

If you need to import other types of tests you should use the [Jira CSV importer](#) instead.

## Prerequisites

For these examples we will use the [Test Case Importer](#) tool, which is present when you install Xray in your Jira instance.

We will need:

- Access to a Jira instance with Xray installed

## Examples

In each example we will have available the CSV file corresponding to the test cases definition and the configuration file used in the Test Case Importer to perform the operation.

We will showcase different possibilities when importing test cases into Xray, using combination of fields and possibilities that should cover most usages.

If you have doubts on where to find the tool or use it please check more information in the [Test Case Importer](#) page.

All of the examples in this tutorial have a CSV file with the definition of the test cases to import and a configuration file that will configure all associations and fields for the importation to be successful.

## Import manual test cases

In this first example we exemplify how to import two test cases:

- Each test case with a proper summary, priority, component association and where we define the manual steps with action, data and result.

The CSV file used is the one below:

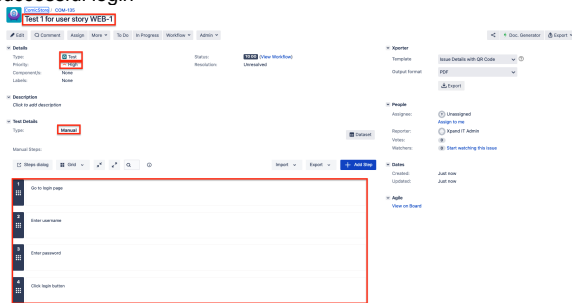
#### import\_two\_testcases.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to login page;;
1;;;;Enter username;peter;
1;;;;Enter password;pwr123;
1;;;;Click login button;;User successfully logged in
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to login page;;
2;;;;Enter username;peter;
2;;;;Enter password;abc123;
2;;;;Click login button;;User logged in fails with message
```

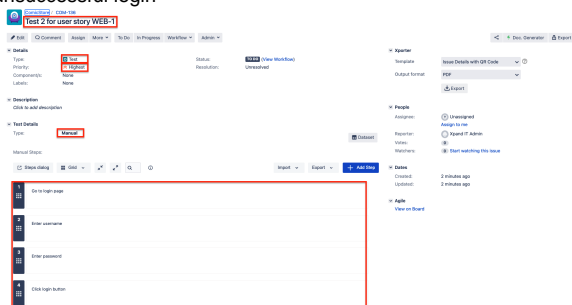
Notice that *Component* and *Priority* fields must match what you have defined in your project otherwise the import will fail.

Once imported we will have two new test cases available:

- one for successful login



- one for unsuccessful login



## Import manual test cases with description

In this example we are importing two manual tests cases with a description.

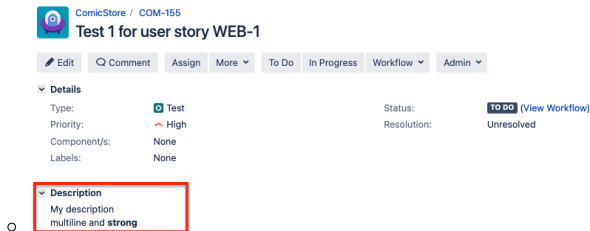
A *Description* field was added to add the information, this field accept Wiki Markup.

#### import\_two\_testcases\_description.csv

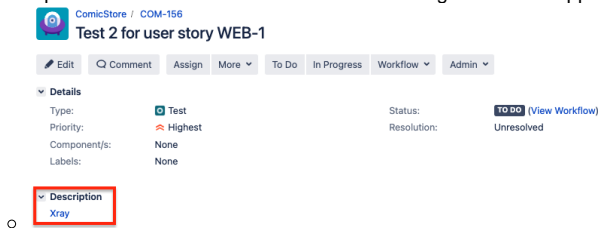
```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Description
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to
login page;;; "My description \\\n multiline and *strong*"
1;;;Enter username;peter;;
1;;;Enter password;pwr123;;
1;;;Click login button;;User successfully logged in;
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to
login page;;; "[Xray|http://www.getxray.app]"
2;;;Enter username;peter;;
2;;;Enter password;abc123;;
2;;;Click login button;;User logged in fails with message;
```

As we can from the CSV above we have added two different descriptions that after the import will appear in Jira:

- The description added to the first test where we can see that is a multiline text with the word **strong** in bold.



- The description of the second test where we are adding a link to an application.



## Import manual test cases with labels

One common case is also the usage of labels, in the following example we are inserting new fields (as much as the number of labels we want to add) called *Label*.

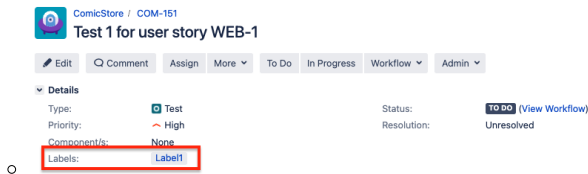
We will see that after importing these test cases they will have associated the labels added in this CSV.

#### import\_two\_testcases\_labels.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Label;Label
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to
login page;;;Label1;
1;;;Enter username;peter;;
1;;;Enter password;pwr123;;
1;;;Click login button;;User successfully logged in;;
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to
login page;;;Label2;Label3
2;;;Enter username;peter;;
2;;;Enter password;abc123;;
2;;;Click login button;;User logged in fails with message;;
```

When importing these test cases we have:

- one test case with *Label1*



- a second test case with two labels: *Label2* and *Label3*



## Import manual test cases with Test Repository

In this example we are using the same base as the previous importation but adding a new field called: *Test Repo Path*, that will allow us to organize these tests into a specific folder in the Test Repository.

### import\_two\_test\_cases\_repo.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Test Repo Path
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to
login page;;Authentication/Login
1;;;Enter username;peter;;
1;;;Enter password;pwr123;;
1;;;Click login button;;User successfully logged in;
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to
login page;;Authentication/Login
2;;;Enter username;peter;;
2;;;Enter password;abc123;;
2;;;Click login button;;User logged in fails with message;
```

On the setup stage make sure you check the option *Create Folders* (as shown in the below screenshot) to create the Test Repository folders in case they do not exist.

**Bulk Test Case Importer Setup**

Setup Project Fields Values

**Setup**

CSV Source File **import\_two\_testcases\_test\_repo.txt**

CSV Delimiter **;** CSV delimiter token

List Delimiter **,** List delimiter token

File Encoding **UTF-8**

**⚠ The CSV file you uploaded contains non-ASCII characters. Make sure to set the right encoding.**

**Hierarchical Test Organization**

☒ **Create Folders**  
 By selecting this option, the imported Test(s) will be organized into new folders, whenever the specified folder path does not exists in the Test Repository.

☒ **Use an existing configuration file**  
 If you have used this importer before, you may have saved the configuration you used. You can use that configuration again to save time.

Configuration File **Choose file** **import\_two...test\_repo.txt**  
 Selecting a configuration file will import the configuration settings. You can still choose to edit these settings further if required. The maximum file size is 10.00 MB.

**ⓘ Please note: Only Tests of Test Type Manual will be imported.**

**Next** **Back**

Once imported the tests will be created, all within the folder *Authentication/Login* of the Test Repository:



## Import manual test cases with Test Repository and link to requirement

When importing manual test cases you can organize them in specific folder within the Test Repository but also link those to the requirements they are covering.

In the CSV file we are using a new field called *requirement* to specify the Issue Key of the requirement we want to link with the test case. In our case we are linking those to the User Story *COM-126*.

### import\_two\_test\_cases\_test\_repo\_link.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Test Repo Path;Requirement
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to
login page;;;Authentication/Login;COM-126
1;;;;Enter username;peter;;;
1;;;;Enter password;pwr123;;;
1;;;;Click login button;;User successfully logged in;;
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to
login page;;;Authentication/Login;COM-126
2;;;;Enter username;peter;;;
2;;;;Enter password;abc123;;;
2;;;;Click login button;;User logged in fails with message;;
```

Notice that you must use valid requirement Keys (that exists in your Jira installation) otherwise the import will fail.

Once the import operation is successful we will have two new manual test cases covering the requirement *COM-126*.

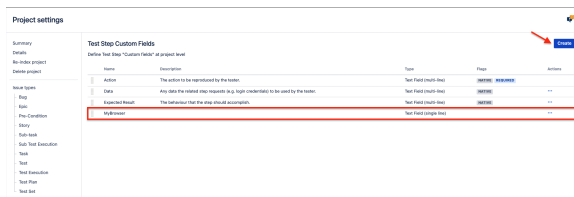


In order to have this example working for you you must replace the requirement Key (COM-126) by requirement Keys that exists in your Jira instance.

## Import manual test cases with custom Test Step and Components

In the next example we are importing two manual test cases and add information into a test step custom field we have added previously and of Components.

The custom Test Step field added is called *MyBrowser*:



As we can see in the contents of the below file, we are using multiple *Component* entries because we want to associate multiple Components to the test case (this will multiply for the amount of Components we want to associate).

We are also adding a *Browser* field that will have the value to be set for that custom field.

### import\_two\_testcases\_custom\_test\_step\_component.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Browser
1;Test 1 for user story WEB-1;High;Interface;Account;Go to login page;;;
Firefox
1;;;Enter username;peter;;
1;;;Enter password;pwd123;;
1;;;Click login button;;User successfully logged in;
2;Test 2 for user story WEB-1;Highest;Interface;Account;Go to login page;;;
Chrome
2;;;Enter username;peter;;
2;;;Enter password;abc123;;
2;;;Click login button;;User logged in fails with message;
```

After importing the CSV file we will have:

- Two new Tests

ComicStore / COM-143

**Test 1 for user story WEB-1**

Edit Comment Assign More To Do In Progress Workflow Admin

**Details**

Type: Test Status: TO DO (View Workflow)

Priority: High Resolution: Unresolved

Component(s): Account, Interface

Labels: None

**Description**

**Test Details**

Type: Manual Dataset

Manual Steps: Click to edit

Steps dialog Grid Q Add Step

Action	Data	Expected Result
Go to login page	None	None
MyBrowser		
Firefox		

Attachments (0)

ComicStore / COM-144

**Test 2 for user story WEB-1**

Edit Comment Assign More To Do In Progress Workflow Admin

**Details**

Type: Test Status: TO DO (View Workflow)

Priority: High Resolution: Unresolved

Component(s): Account, Interface

Labels: None

**Description**

**Test Details**

Type: Manual Dataset

Manual Steps:

Steps dialog Grid Q Add Step

Action	Data	Expected Result
Go to login page	None	None
MyBrowser		
Chrome		

Attachments (0)

Enter username

Enter password

## Import manual test cases with attachments

In this example we will import two manual Tests with *Attachments* (either in the test step or in the test case).

For that we have added three new fields:

- Attach Steps - That will have the attachments to be added in the Test Steps;
- Attach - To add attachments in the test case.

The attachment may be referred by name (in this case the file:// protocol is assumed), by "file://" or "http(s)://". Multiple files may be specified within one CSV column; for this, the attachments must be delimited by space.

For file protocol, only files in the "<JIRA\_DATA\_HOME>/import/attachments" folder will be imported.

Looking at the CSV file we can see those new fields:

## import\_two\_testcases\_comments\_attachments.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Attach Steps;Attach
1;Test 1 for user story WEB-1;High;User Interface;Business Layer;Go to
login page;;;https://docs.getxray.app/download/attachments/37062075
/image2022-2-15_18-9-33.png
1;;;Enter username;peter;;;
1;;;Enter password;pwr123;;;
1;;;Click login button;;User successfully logged in;;
2;Test 2 for user story WEB-1;Highest;User Interface;Business Layer;Go to
login page;;;
2;;;Enter username;peter;;;https://docs.getxray.app/download/attachments
/37062075/image2022-2-15_18-9-33.png;
2;;;Enter password;abc123;;;https://docs.getxray.app/download/attachments
/37062075/image2022-2-15_18-9-33.png;
2;;;Click login button;;User logged in fails with message;;https://docs.
getxray.app/download/attachments/37062075/image2022-2-15_18-9-33.png;
```

This will create two Tests, one Test Case with an attachment in the test step and another with an attachment in the Test as we can see:

The screenshot displays two Jira test case pages. The top page is for 'Test 1 for user story WEB-1', which is a 'Test' type with a 'High' priority. It shows an attachment 'Image2022-2-15\_18-9-33' in the 'Attachments' section. The bottom page is for 'Test 2 for user story WEB-1', which is a 'Manual' type with a 'Highest' priority. It shows a 'Manual Steps' table with two steps: 'Go to login page' and 'Enter username'. Both steps have an attachment 'Image2022-2-15\_18-9-33' attached to them. Red boxes and arrows highlight the attachment links in both test cases.

**Test 1 for user story WEB-1**

Type: Test  
Priority: High  
Component(s): None  
Labels: None  
Status: Unresolved  
Resolution: Unresolved

**Test 2 for user story WEB-1**

Type: Manual  
Priority: Highest  
Component(s): None  
Labels: None  
Status: Unresolved  
Resolution: Unresolved

Manual Steps:

Step	Action	Data	Expected Result
1	Go to login page		
2	Enter username	peter	None
3	Enter password	abc123	None

Please check this [page](#) for more details on the attachment features.

## Mapping priorities



In some cases we can use priorities in our CSV file that do not match the ones we have defined in the Jira project (using different definitions or just different lower/upper case in the word).

Instead of changing all priority fields in all CSV files we can define a mapping that will associate the priorities in our CSV file to the ones in the Jira project.

We will use the below CSV file to demonstrate this functionality.

#### import\_two\_testcases\_mapping\_priority.csv

```
TCID;Test Summary;Test Priority;Component;Component;Action;Data;Result
1;Test 1 for user story WEB-1;high;User Interface;Business Layer;Go to login page;;
1;;;Enter username;peter;
1;;;Enter password;pwr123;
1;;;Click login button;;User successfully logged in
2;Test 2 for user story WEB-1;low;User Interface;Business Layer;Go to login page;;
2;;;Enter username;peter;
2;;;Enter password;abc123;
2;;;Click login button;;User logged in fails with message
```

To accomplish that we will check the checkbox *Map Field Value* next to the *Test Priority* field:

The screenshot shows the 'Fields' configuration page in Jira. The 'Test Priority' field is selected, and the 'Map Field Value' checkbox is checked, indicated by a red arrow.

External Field	Xray Field	Map Field Value
TCID (e.g. 1)	Test Case Identifier	<input type="checkbox"/>
Test Summary (e.g. Test 1 for user story WEB-1)	Summary	<input type="checkbox"/>
Test Priority (e.g. high)	Priority	<input checked="" type="checkbox"/>
Component (e.g. User Interface)	Component/s	<input type="checkbox"/>
Action (e.g. Go to login page)	Action	<input type="checkbox"/>
Data (e.g. peter)	Data	<input type="checkbox"/>
Result (e.g. User successfully logged in)	Expected Result	<input type="checkbox"/>

Next we will define the mapping between our priorities and the ones present in the Jira project:

The screenshot shows the 'Map values' configuration page in Jira. The 'Test Priority' field is selected, and the mapping between 'high' and 'Low' is shown.

CSV Field	Value from Importer	Target value in Jira
Test Priority (Imported as Priority)	high	High
	low	Low

Finally we will see that the imported Tests will have the correct association of priorities:

ComicStore / COM-147

### Test 1 for user story WEB-1

Edit Comment Assign More To Do In Progress Workflow Admin

**Details**

Type: Test  
 Priority: High  
 Component/s: None  
 Labels: None

Status: **TO DO** (View Workflow)  
 Resolution: Unresolved

ComicStore / COM-148

### Test 2 for user story WEB-1

Edit Comment Assign More To Do In Progress Workflow Admin

**Details**

Type: Test  
 Priority: Low  
 Component/s: None  
 Labels: None

Status: **TO DO** (View Workflow)  
 Resolution: Unresolved

## Update two manual test cases

If your intent is not to create new test cases but to update test cases you must refer to the test case Issue Key that you want to update.

In the setup steps we will need to map the fields to be used, make sure you are mapping the *Key* to the *Issue Key* so that the update to happen.

**Fields** Setup Project Fields Values

**Fields**

*You have selected Issue Key mapping. Please note that importer will update existing issues matching the Issue Key from the CSV file.*

External Field	Xray Field	Map Field Value
TCID (e.g. 1)	Test Case Identifier	<input type="checkbox"/>
<b>Key</b> (e.g. COM-147)	Issue Key	<input type="checkbox"/>
Test Summary (e.g. Test 1 for user story WEB-1 Up ...)	Summary	<input type="checkbox"/>
Test Priority (e.g. High)	Priority	<input type="checkbox"/>
Component (e.g. User Interface)	Component/s	<input type="checkbox"/>
Action (e.g. Go to login page)	Action	<input type="checkbox"/>
Data (e.g. peter)	Data	<input type="checkbox"/>
Result (e.g. User successfully logged in)	Expected Result	<input type="checkbox"/>
Test Repo Path (e.g. Authentication/Login)	Test Repository Path	<input type="checkbox"/>
Requirement (e.g. COM-126)	Link "Tests"	<input type="checkbox"/>

**Next** Back

When importing the Test Case Importer will update the test case instead of creating a new.

To exemplify I have created the following CSV file

### update\_two\_testcases\_test\_repo\_link.csv

```
TCID;Key;Test Summary;Test Priority;Component;Component;Action;Data;Result;
Test Repo Path;Requirement
1;COM-147;Test 1 for user story WEB-1 Updated;High;User Interface;Business
Layer;Go to login page;;;Authentication/Login;COM-126
1;;;;;Enter username;peter;;;
1;;;;;Enter password;pwr123;;;
1;;;;;Click login button;;User successfully logged in;;
2;COM-148;Test 2 for user story WEB-1 Updated;Highest;User Interface;
Business Layer;Go to login page;;;Authentication/Login;COM-126
2;;;;;Enter username;peter;;;
2;;;;;Enter password;abc123;;;
2;;;;;Click login button;;User logged in fails with message;;
```

With this import we will update those two test cases (*COM-147* and *COM-148*).

The image shows two screenshots of Jira test case details. The top screenshot is for test case 'COM-147' titled 'Test 1 for user story WEB-1 Updated'. It shows details like Type: Test, Priority: High, Component: None, and Test Repository Path: Authentication/Login. The bottom screenshot is for test case 'COM-148' titled 'Test 2 for user story WEB-1 Updated'. It shows details like Type: Test, Priority: Highest, Component: None, and Test Repository Path: Authentication/Login. Both screenshots show the 'Manual' test type and a single step: 'Go to login page'.

## Tips

- Use the CSV file provided to import the examples and the txt configuration associated to it in order to have the correct setup and mapping.
- Remind to replace the Project Keys, test case Keys, Custom Fields and requirements for some existing in your Jira instance.