


Importing TestRail test cases using Test Case Importer

 What you'll learn

- [Features and Limitations](#)
- [How it works](#)
- [Prerequisites](#)
- [Examples](#)
 - [Export XML from TestRail](#)
 - [Validate that the test cases were imported to Xray](#)
 - [More Details](#)

- [Test Sections into Test Repository](#)
- [Manual Tests \(With Steps separated and in Text\) into Manual Test Case](#)
- [Other Tests](#)

Source code for this tutorial

[Types](#)

- All examples are available in [GitHub](#)

Overview

This tutorial explains how to export test cases from TestRail and import them into Xray.

TestRail is a test management tool that runs independently with versions available for on-premises and in the cloud. In this tutorial, you will learn how to export the test cases from TestRail in an XML format, how to use a script (available in [GitHub](#)) to convert the XML file into a compatible CSV so that it can be imported into Xray using the [Test Case Importer](#).

In this tutorial, we will provide different examples of exporting and importing test cases from TestRail into Xray.

Features and Limitations

Below you can find a list of the support features and current limitations.

Migration from TestRail is limited the information we are able to export from the TestRail and how it can be mapped to Xray entities.

Most of the test case information will be migrated seamlessly, but please check the following table for more detail.

Supported Features	Unsupported
<ul style="list-style-type: none">• Test and Test steps• Test Sections/SubSections	<ul style="list-style-type: none">• Test case attachments• Preconditions• Test Runs• Test Plans• Defects and Requirements

How it works

TestRail allows users to create different test projects. The user can specify and organize those tests into test sections. Each test project can have multiple test sections and subsections.

TestRail has a set of templates that we choose upon test case creation that will determine what information is present in the test. The default templates include the "Test Case (Text)" template that allows you to describe the test case in a text format; we also have a "Test Case (Steps)" that allows you to describe a test case in different steps with action, additional data and expected result. TestRail also has a Behaviour Driven Development template that enables the user to describe the test into different scenarios or an Exploratory test type that defines a mission and goals for the test.

In TestRail, to execute tests you need to associate them to a test run.

We have summarised a possible mapping between TestRail and Xray entities that we'll use in this tutorial.

TestRail	Xray	Comments
Test Case	Test	TestRail does not support data sets or Gherkin definition, only manual /automated test steps. In TestRail when you choose to export using XML format it will export everything. All steps are exported unless they are empty.
Test Section	Test Repository	The test sections and sub-sections of TestRail are mapped into Xray Test Repository.



Notice that for the Server version of Xray we only creating the Tests corresponding to the manual test cases in TestRail. Preconditions, if needed, must be created separately.

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 TestRail instance to export test cases in XML format
- Usage of scripts to convert TestRail XML test cases to CSV Xray import files
- Access to a Jira instance with Xray installed

Examples

In each example, we will show how we obtained the XML file from TestRail, how to use the scripts to convert the XML files into compatible CSV files, and finally, how to import the CSV files into Xray. All files are available in [GitHub](#).

We will showcase different possibilities when exporting test cases from TestRail, using a combination of fields and possibilities that should cover most usages.

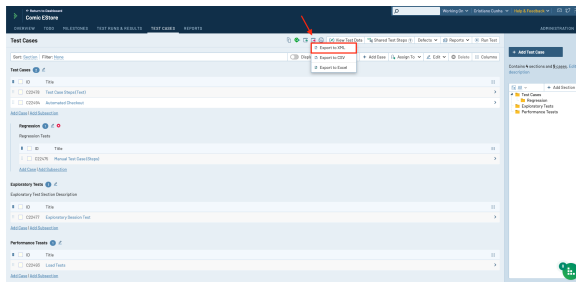
All of the examples in this tutorial have a XML file exported from TestRail, a CSV file with the definition of the test cases to import, and a configuration file that will configure all associations and fields for the importation to be successful in Xray.

Export to XML from TestRail and Import into Xray

In TestRail when you choose to export using XML format it will export everything, no option is given to choose to specify what we want to export. In this example we are focusing on that option.

To export from TestRail we are using the *Export to XML* option in the Test Case upper menu entry.

We have to provide a name for the XML field that will be exported and click "*Export*".



Looking into more detail on what we have in TestRail, we can see that we have defined 3 sections:

- Test Cases (with one sub-section)
 - Regression
- Exploratory Tests
- Performance Tests

The file generated by TestRail has the following content:

comic_estore.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<suite>
  <id>S196</id>
  <name>Master</name>
  <description></description>
  <sections>
    <section>
      <name>Test Cases</name>
      <description></description>
      <cases>
        <case>
          <id>C22478</id>
          <title>Test Case Steps (Text)</title>
          <template>Test Case (Text)</template>
          <type>Destructive</type>
          <priority>Medium</priority>
          <estimate></estimate>
          <references></references>
          <custom>
            <automation_type><id>6</id><value> Manual</value></automation_type>
            <preconds>Test Case Step *TEXT* Precondition</preconds>
            <steps>%Var3
%Var4
</steps>
            <expected>  </expected>
          </custom>
        </case>
        <case>
          <id>C22494</id>
          <title>Automated Checkout</title>
          <template>functional area</template>
```

```

<type>Automated</type>

<priority>Critical</priority>

<estimate></estimate>

<references></references>

</case>

</cases>
<sections>
    <section>
        <name>Regression</name>
        <description>Regression Tests</description>
    </section>
</cases>

<case>
    <id>C22475</id>
    <title>Manual Test Case (Steps)</title>

    <template>Test Case (Steps)</template>

    <type>UAT</type>

    <priority>Medium</priority>

    <estimate></estimate>

    <references></references>

    <custom>

    <automation_type><id>6</id><value> Manual</value></automation_type>

    <preconds>Manual Test Case (Steps) Precondition</preconds>

    <steps_separated><step>
        <index>1</index><content>Login with user *%User1*
        Expected Result:
        Application redirects you to the main page in a logged in state</content><
    /step>
    <step>
        <index>2</index><content>First step description<
        /content><expected>Expected result for the first step<
        /expected><additional_info>Additional step description</additional_info><
    /step>
    <step>
        <index>3</index><content>Second step description with two variables: %
        Var1% and %Var2% </content><expected>Expected result for the second step<
        /expected><additional_info>Additional step 2 description</additional_info><
    /step>
    <step>
        <index>4</index><content>Third step with a table:

        |||:Name 1|:Last name 2|:Phone 3|:Address 4
        || John .. | Smack | 111111111 | Address1
        || Smith .. | Moon | 222222222 | Address2
        || Jenkins .. | Stone | 333333333 | Address3
        || Ford .. | Mustang | 444444444 | Address4
        </content><expected>Expected result for the third step<
        /expected><additional_info>Additional step 2 description (Var1: %Var1% and
        Var2: %Var2%)</additional_info></step>
    <step>
        <index>5</index><content>  </content><expected>  </expected><additional_info> ![]
        (index.php?/attachments/get/5edccf31-330d-4e62-9e8a-324f771d96c8) <

```

```

/additional_info></step>
</steps_separated>

/custom>
</case>
</cases>
</section>
</sections>
</section>
<section>
<name>Exploratory Tests</name>
<description>Exploratory Test Section Description<
/description>
<cases>
<case>
<id>C22477</id>
<title>Exploratory Session Test</title>

<template>Exploratory Session</template>
<type>Regression</type>
<priority>Medium</priority>
<assignedto>Cristiano Cunha</assignedto>
<estimate></estimate>
<references></references>
<custom>
<mission>Exploratory Mission</mission>
<goals>Exploratory Goal</goals>
</custom>
</case>
</cases>
</section>
<section>
<name>Performance Tests</name>
<description></description>
<cases>
<case>
<id>C22493</id>
<title>Load Tests</title>

<template>functional area</template>
<type>Performance</type>
<priority>Medium</priority>
<estimate></estimate>
<references></references>
</case>
</cases>
</section>
</sections>
</suite>

```

To import into Xray, we created one script to convert this XML file into a CSV-compatible file to be imported in Xray.

The script will parse the XML file and create a CSV file that can be imported into Xray. To execute the script, use the following command:

```
python3 testrail2Xray.py -i comic_estore.xml -o comic_estore.csv
```

An optional parameter (-e) is also available to complete the endpoints sent in the XML file. To just append the above command with the -e option and sent the complete endpoint of your TestRail instance. For example if my TestRail instance is "https://mytestrail.com/" the command will look like this:

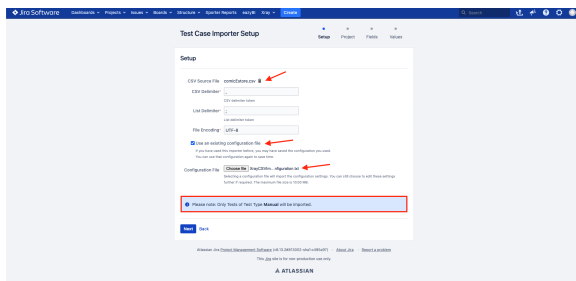
```
python3 testrail2Xray.py -i comic_estore.xml -o comic_estore.csv -e  
https://mytestrail.com/
```

The output file will only have the Manual Tests, for those all the test steps will be present.

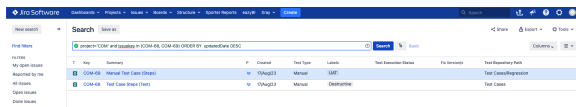
comic_estore.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,  
Test Repo,Labels  
1,,Manual,Test Case Steps (Text),3,"%Var3  
%Var4  
",,  ,  
Test Cases,Destructive  
2,,Manual,Manual Test Case (Steps),3,"Login with user *%User1*  
Expected Result:  
Application redirects you to the main page in a logged in state",,,Test  
Cases/Regression,UAT  
2,,Manual,,3,First step description,Additional step description,Expected  
result for the first step,,  
2,,Manual,,3,Second step description with two variables: %Var1% and %Var2%  
,Additional step 2 description,Expected result for the second step,,  
2,,Manual,,3,"Third step with a table:  
  
||:Name 1|:Last name 2|:Phone 3|:Address 4  
|| John .. | Smack | 111111111 | Address1  
|| Smith .. | Moon | 222222222 | Address2  
|| Jenkins .. | Stone | 333333333 | Address3  
|| Ford .. | Mustang | 444444444 | Address4  
",Additional step 2 description (Var1: %Var1% and Var2: %Var2%),Expected  
result for the third step,,  
2,,Manual,,3,  ,  ,  ,,
```

To import it into Xray we use Test Case Importer feature of Xray with the recently created CSV file; you can use the configuration file provided with the txt extension (*XrayCSVImporter-configuration.txt*).



Once imported we can see that it has created two Tests with the properties defined in the CSV.

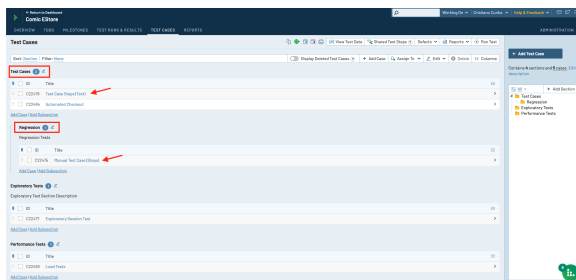


More Details

Let's look into some specificities of this approach and some test particularities.

Test Sections into Test Repository

In TestRail we can organize our tests into sections and subsections as we can see below:



As we can see in the above picture we have several sections but only the ones with manual tests will be created in Xray. The sections we have in TestRail are:

- one section called 'Test Cases' with 2 tests and a sub-section 'Regression' with one test
- one section called 'Exploratory Tests' with one test
- one section called 'Performance Tests' with one test

TestRail sections and sub-sections are exported into the XML file in the following form:

test section in comic_estore.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<suite>
  <id>S196</id>
  <name>Master</name>
  <description></description>
  <sections>
    <section>
      <name>Test Cases</name>
      <description></description>
      ...
      <sections>
        <section>
          <name>Regression</name>
          <description>Regression Tests<
/description>
          ...
        </section>
      </sections>
    </section>
    <section>
      <name>Exploratory Tests</name>
      <description>Exploratory Test Section Description<
/description>
      <cases>
        ...
      </cases>
    </section>
    <section>
      <name>Performance Tests</name>
      <description></description>
      <cases>
        <case>
          ...
        </case>
      </cases>
    </section>
  </sections>
</suite>
```

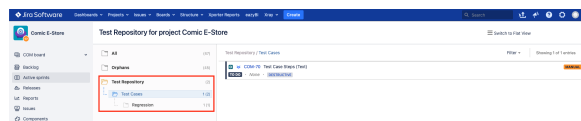
After running the script that converts the XML file into a CSV format we can see that it adds, in the *Test Repo field*, the Test Repository name to be created, only for the manual tests, when uploading the file into Xray.

Test Repo in comic_estore.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,
Test Repo,Labels
1,,Manual,Test Case Steps (Text),3,"%Var3
%Var4
",, [Link|https://testrail101.testrail.io/index.php?/attachments/get
/d8d1d9e0-fe4c-43c8-b894-dfd83d9c160c] ,Test Cases,Destructive
2,,Manual,Manual Test Case (Steps),3,"Login with user *%User1*
Expected Result:
Application redirects you to the main page in a logged in state",,,Test
Cases/Regression,UAT
2,,Manual,,3,First step description,Additional step description,Expected
result for the first step,,
2,,Manual,,3,Second step description with two variables: %Var1% and %Var2%
,Additional step 2 description,Expected result for the second step,,
2,,Manual,,3,"Third step with a table:

||:Name 1|:Last name 2|:Phone 3|:Address 4
| John .. | Smack | 111111111 | Address1
| Smith .. | Moon | 222222222 | Address2
| Jenkins .. | Stone | 333333333 | Address3
| Ford .. | Mustang | 444444444 | Address4
",Additional step 2 description (Var1: %Var1% and Var2: %Var2%),Expected
result for the third step,,
2,,Manual,,3, [Link|https://testrail101.testrail.io/index.php?/attachments
/get/3b93e38b-5609-49c6-a987-8f2f6968121e] , [Link|https://testrail101.
testrail.io/index.php?/attachments/get/5edccf31-330d-4e62-9e8a-
324f771d96c8] , [Link|https://testrail101.testrail.io/index.php?
/attachments/get/6c9ccdd6-446f-4b23-9244-3a16b997a0e6] , ,
```

After importing into Xray we can see that new directories were created in the Test Repository:



Manual Tests (With Steps separated and in Text) into Manual Test Case

The templates available by default in TestRail allow you to create Manual Test Cases with steps (using action, data and expected result) or using only a text field to describe the steps.

We have created one test case using the steps in a text field and one using the steps separated into fields, the XML output is the following:

Manual Tests in comic_estore.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<suite>
  <id>S196</id>
  <name>Master</name>
  <description></description>
  <sections>
    <section>
      <name>Test Cases</name>
      <description></description>
      <cases>
        <case>
          <id>C22478</id>
```

```

<title>Test Case Steps (Text)<
/title>
<template>Test Case (Text)<
<type>Destructive</type>
<priority>Medium</priority>
<estimate></estimate>
<references></references>
<custom>
<automation_type><id>6<
/id><value> Manual</value></automation_type>
<preconds>Test Case Step
*TEXT* Precondition</preconds>
<steps>%Var3
%Var4
</steps>
<expected>  </expected>
</custom>
</case>
...
</cases>
<sections>
<section>
<name>Regression</name>
<description>Regression Tests</description>
<cases>
<case>
<id>C22475</id>
<title>Manual Test Case (Steps)<
/title>
<template>Test Case (Steps)<
<type>UAT</type>
<priority>Medium</priority>
<estimate></estimate>
<references></references>
<custom>
<automation_type><id>6<
/id><value> Manual</value></automation_type>
<preconds>Manual Test Case
(Steps) Precondition</preconds>
<steps_separated>
<step>
<index>1<
/index>
<content>Login with user *%User1*
Expected Result:
Application redirects you to the main page in a logged in state</content>
</step>
<step>
<index>2<
/index>
<content>First step description</content>
<expected>Expected result for the first step</expected>
<additional_info>Additional step description</additional_info>
</step>
<step>
<index>3<
/index>
<content>Second step description with two variables: %Var1% and %Var2% <
/content>
<expected>Expected result for the second step</expected>

```

```

<additional_info>Additional step 2 description</additional_info>
                                </step>
                                <step>
                                    <index>4<
/idx>

<content>Third step with a table:

||:Name 1|:Last name 2|:Phone 3|:Address 4
|| John .. | Smack | 111111111 | Address1
|| Smith .. | Moon | 222222222 | Address2
|| Jenkins .. | Stone | 333333333 | Address3
|| Ford .. | Mustang | 444444444 | Address4

                                </content>

<expected>Expected result for the third step</expected>

<additional_info>Additional step 2 description (Var1: %Var1% and Var2: %
Var2%)</additional_info>
                                </step>
                                <step>
                                    <index>5<
/idx>

                                <content> !
[ ](index.php?/attachments/get/3b93e38b-5609-49c6-a987-8f2f6968121e) <
/content>

                                <expected>
![ ](index.php?/attachments/get/6c9ccdd6-446f-4b23-9244-3a16b997a0e6) <
/expected>

<additional_info> ![ ](index.php?/attachments/get/5edccf31-330d-4e62-9e8a-
324f771d96c8) </additional_info>
                                </step>
                                </steps_separated>
                                </custom>
                                </case>
                                </cases>
                                </section>
                                </sections>
                                ...

```

After running the script that converts this XML into a CSV file we can see that, each Test is created and the steps filled with the information extracted from the XML.

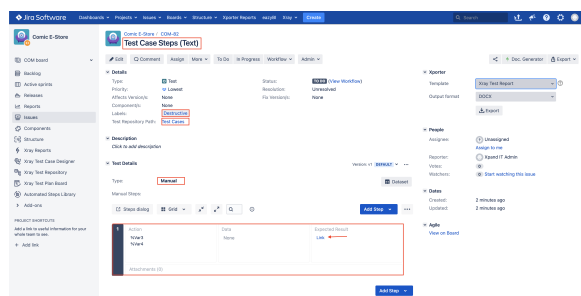
Manual Tests in comic_estore.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,
Test Repo,Labels
1,,Manual,Test Case Steps (Text),3,"%Var3
%Var4
",,,[Link|https://testrail101.testrail.io/index.php?/attachments/get
/d8d1d9e0-fe4c-43c8-b894-dfd83d9c160c],Test Cases,Destructive
2,,Manual,Manual Test Case (Steps),3,"Login with user *%User1*
Expected Result:
Application redirects you to the main page in a logged in state",,,Test
Cases/Regression,UAT
2,,Manual,,3,First step description,Additional step description,Expected
result for the first step,,
2,,Manual,,3,Second step description with two variables: %Var1% and %Var2%
,Additional step 2 description,Expected result for the second step,,
2,,Manual,,3,"Third step with a table:

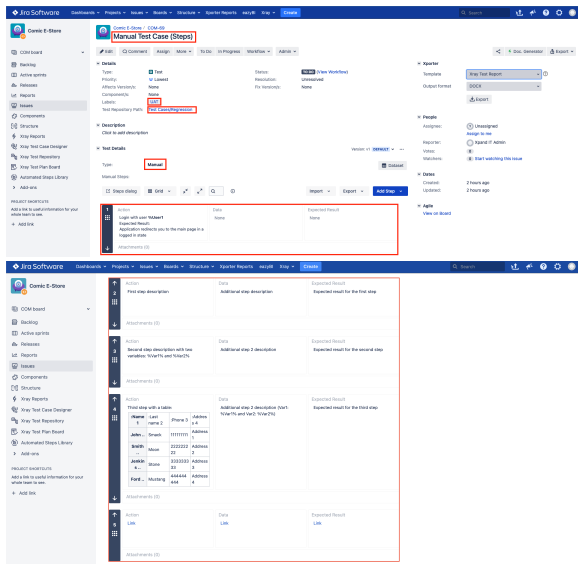
||:Name 1|:Last name 2|:Phone 3|:Address 4
|| John .. | Smack | 111111111 | Address1
|| Smith .. | Moon | 222222222 | Address2
|| Jenkins .. | Stone | 333333333 | Address3
|| Ford .. | Mustang | 444444444 | Address4
",Additional step 2 description (Var1: %Var1% and Var2: %Var2%),Expected
result for the third step,,
2,,Manual,,3, [Link|https://testrail101.testrail.io/index.php?/attachments
/get/3b93e38b-5609-49c6-a987-8f2f6968121e] , [Link|https://testrail101.
testrail.io/index.php?/attachments/get/5edccf31-330d-4e62-9e8a-
324f771d96c8] , [Link|https://testrail101.testrail.io/index.php?
/attachments/get/6c9ccdd6-446f-4b23-9244-3a16b997a0e6] , ,
```

The first Test corresponds to a Manual Test Case with the summary obtained from the CSV file, one step with Action and Expected Result and the added type (*Destructive*) as a Label.

Notice that we have added one image in the Expected Result in TestRail and the exported link is reconstructed using the `-e` option of the script.



The second Test has one line per each step. Notice that the format is kept for the table and styles added in TestRail.



Other Tests

The script will convert all tests of type **Manual**, all the other types will be discarded and no output is created in the resulting CSV file.



The Type field (e.g., "Destructive", "Smoke", "UAT") in TestRail is converted into labels in the Test issues managed by Xray.



BDD Test Cases are not covered in this tutorial due to a limitation in TestRail.

Tips

- Use the CSV file provided to import the examples and the Test Case Importer configuration file associated with it in order to have the setup and mapping correct.
- Make sure you are using the scripts available for the right version of Jira (Cloud or Server).