[Xray] Export from Cloud to Data Center

Xray, at the moment, doesn't have any direct way to transfer information between Cloud and Data Center.

This article will guide users by providing some steps to perform that task manually.

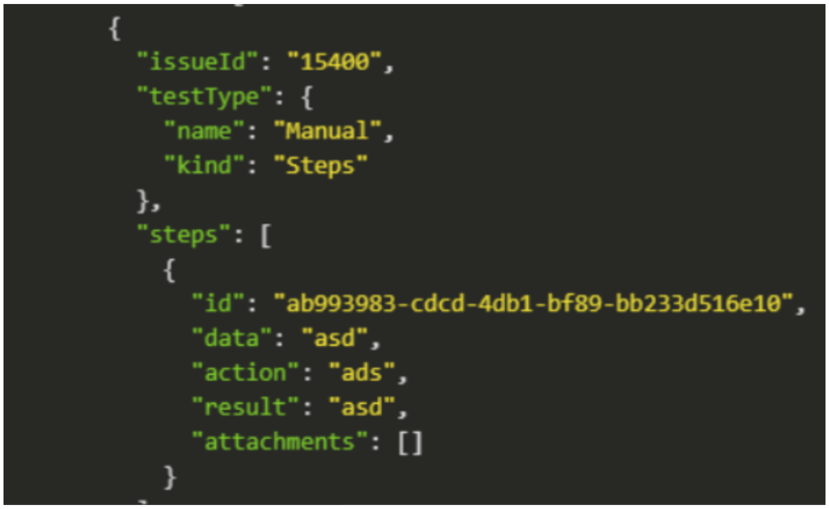
1 - How to export the tests with steps from the Xray cloud

To export the Tests with steps, you can use GraphQL API, with which you can ask exactly what you need and nothing more.
To make a request to Xray's GraphQL API, for starters, you need to be authenticated using the same end for Rest API authentication. As an HTTP client, we use Insomnia, which is native with GraphQL support. After the connections have been made, use our Xray's GraphQL Schema Documentation which provides the queries to get data from Xray.

For this example, we will want to export the Tests with steps, and this section will help you with that, **Get tests**.

2 - How to change the format the Xray Data Center can accept

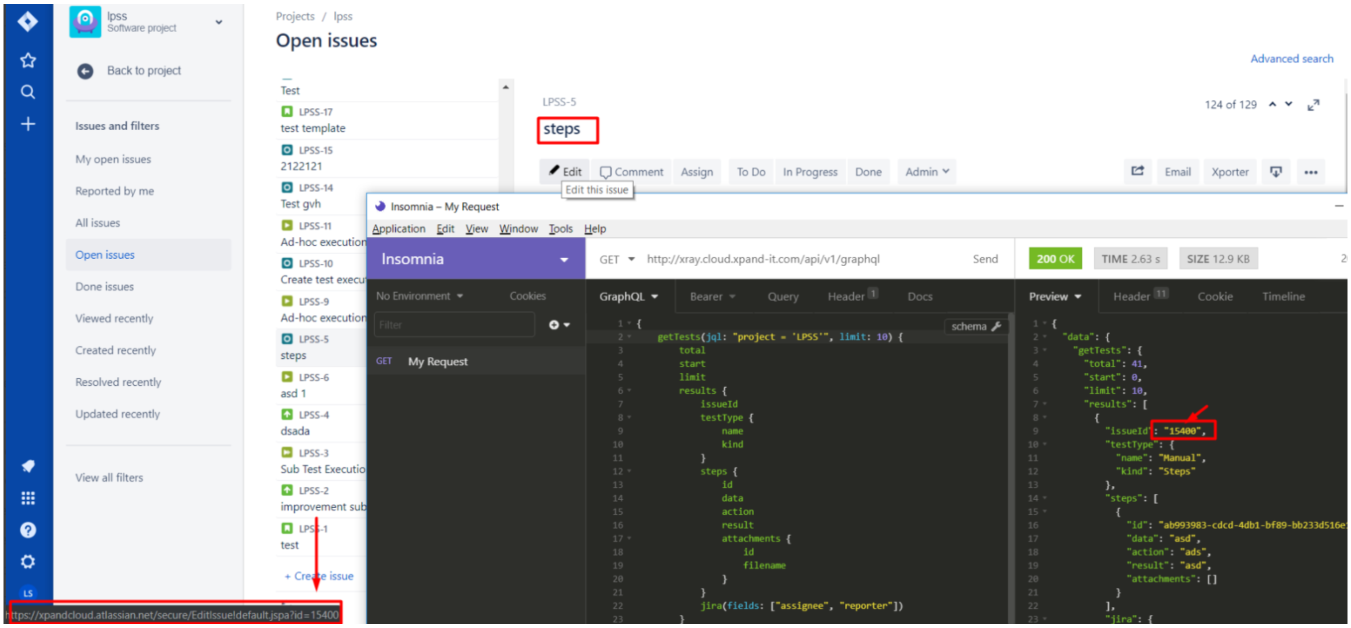
After exporting, on the Insomnia Screen, you will get the information that you requested for the Test and Steps. Between those lines, when you find parts like this:



```
{
  "issueId": "15400",
  "testType": {
    "name": "Manual",
    "kind": "Steps"
  },
  "steps": [
    {
      "id": "ab993983-cdcd-4db1-bf89-bb233d516e10",
      "data": "asd",
      "action": "ads",
      "result": "asd",
      "attachments": []
    }
```

This means this is a Test! So that is the information that will be used for the CSV.

NOTE: The summary of each test will not be exported; it will need to be filled manually by checking each ID and comparing it with the test ID of the result of GraphQL:



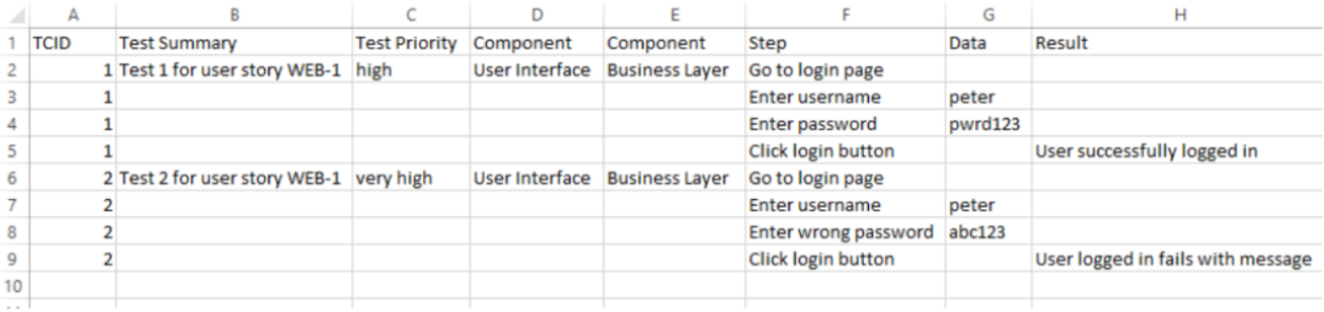
```
{
    getTests(jql: "project = 'LPSS'", limit: 10) {
        total
        start
        limit
        results {
            issueId
            testType {
                name
                kind
            }
            steps {
                id
                data
                action
                result
                attachments {
                    id
                    filename
                }
            }
            jira(fields: ["assignee", "reporter"])
        }
```

```
{
  "data": {
    "getTests": {
      "total": 41,
      "start": 0,
      "limit": 10,
      "results": [
        {
          "issueId": "15400",
          "testType": {
            "name": "Manual",
            "kind": "Steps"
          },
          "steps": [
            {
              "id": "ab993983-cdcd-4db1-bf89-bb233d516e
              "data": "asd",
              "action": "ads",
              "result": "asd",
              "attachments": []
            }
          ],
          "jira": {
```

(Users will need to move the mouse over the edit button, then it will show you on the bottom of the screen the Issue ID)

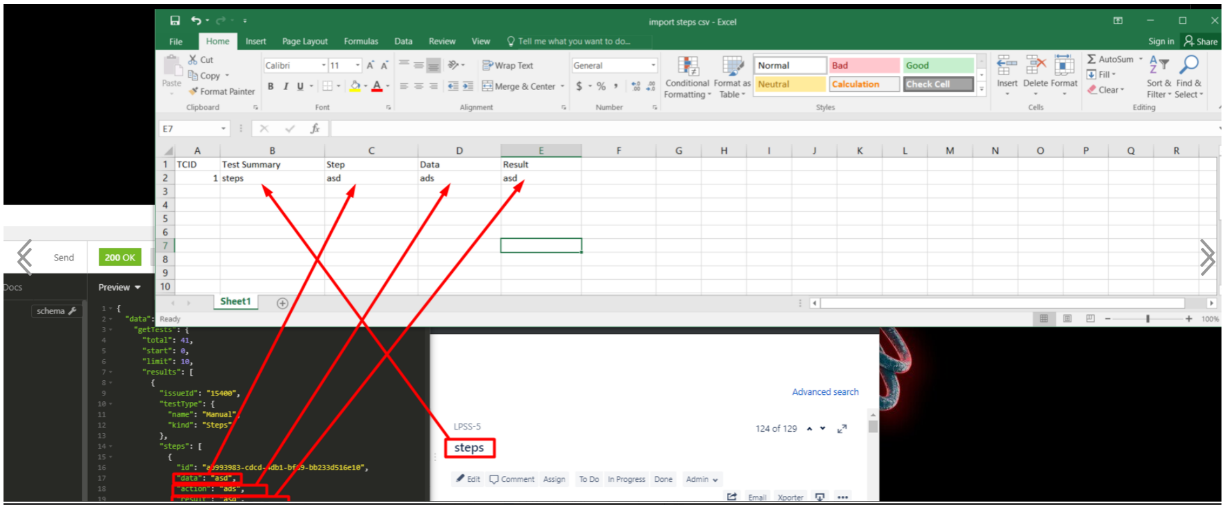
I mention the field summary because it is a field that is needed to have when using the Test Case Importer!

The next step is to add that information to an Excel that meets this test case structure:



	A	B	C	D	E	F	G	H
1	TCID	Test Summary	Test Priority	Component	Component	Step	Data	Result
2	1	Test 1 for user story WEB-1	high	User Interface	Business Layer	Go to login page		
3	1					Enter username	peter	
4	1					Enter password	pwrd123	
5	1					Click login button		User successfully logged in
6	2	Test 2 for user story WEB-1	very high	User Interface	Business Layer	Go to login page		
7	2					Enter username	peter	
8	2					Enter wrong password	abc123	
9	2					Click login button		User logged in fails with message
10								

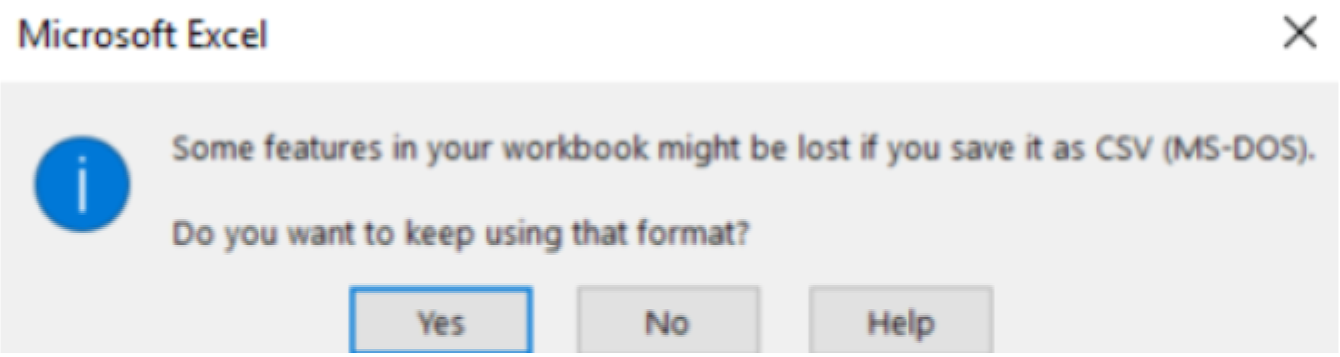
Using the example that I have been giving, it will be something like this:



	A	B	C	D	E
1	TCID	Test Summary	Step	Data	Result
2	1	steps	asd	ads	asd

```
{
  "data": {
    "getTests": {
      "total": 41,
      "start": 0,
      "limit": 10,
      "results": [
        {
          "issueId": "15400",
          "testType": {
            "name": "Manual",
            "kind": "Steps"
          },
          "steps": [
            {
              "id": "a0993983-cdcd-4db1-bf59-bb233d516e10",
              "data": "asd",
              "action": "ads",
              "result": "asd",
```

After this, save it in a CSV format:



Microsoft Excel

Some features in your workbook might be lost if you save it as CSV (MS-DOS).

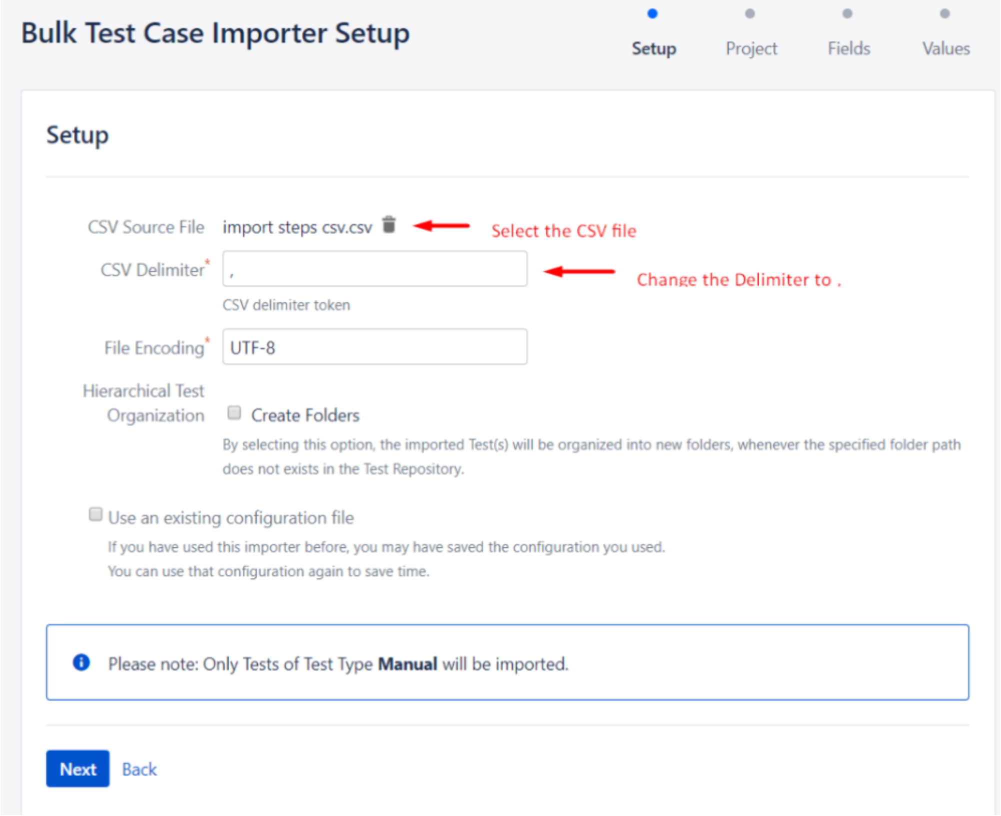
Do you want to keep using that format?

Press "Yes"

- **3- How to import it with the corresponding steps**

When we have the CSV with all the information on the Tests, we can start to Import the data into the Data Center version. To do that, you can use Xray's Test Case Importer (a Jira Importer Plugin extension) to import Manual Tests from a CSV source file.

Let's then go to the Test Case Importer and edit the options:



Bulk Test Case Importer Setup

Setup

CSV Source File: import steps csv.csv

CSV Delimiter*: ,

CSV delimiter token

File Encoding*: UTF-8

Hierarchical Test Organization: □ Create Folders

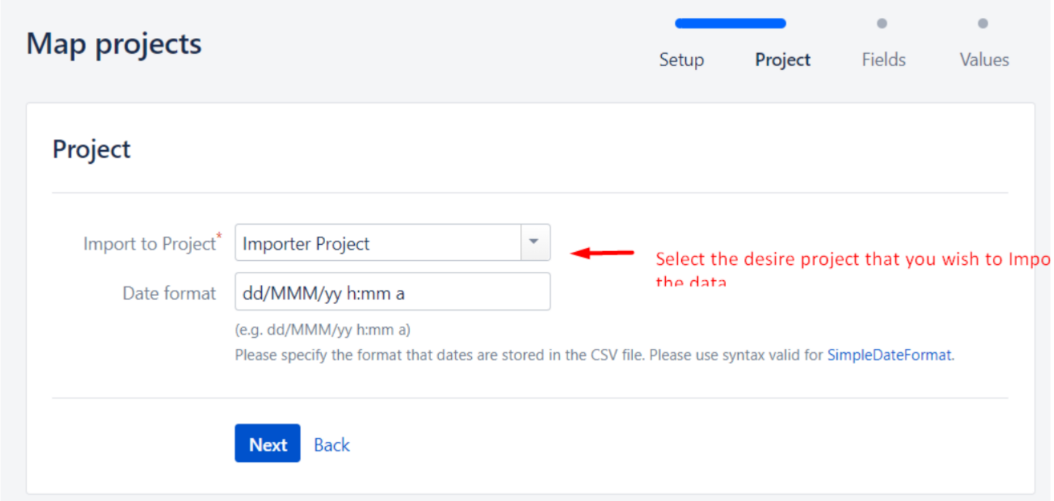
By selecting this option, the imported Test(s) will be organized into new folders, whenever the specified folder path does not exists in the Test Repository.

□ 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.

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

Select the project that you wish to import:



Map projects

Project

Import to Project* Importer Project

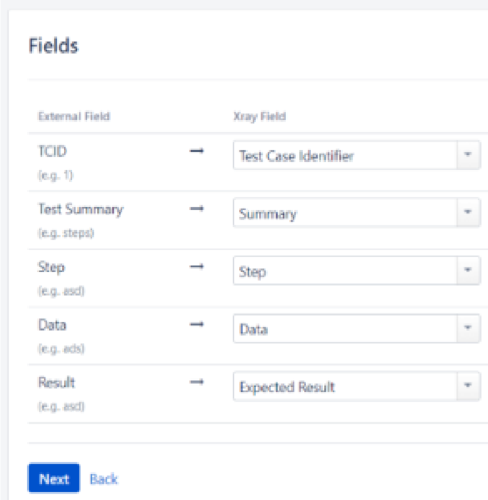
Date format dd/MMM/yy h:mm a

(e.g. dd/MMM/yy h:mm a)

Please specify the format that dates are stored in the CSV file. Please use syntax valid for SimpleDateFormat.

Select the desire project that you wish to Impo the data

Map the fields with fields from the CSV:

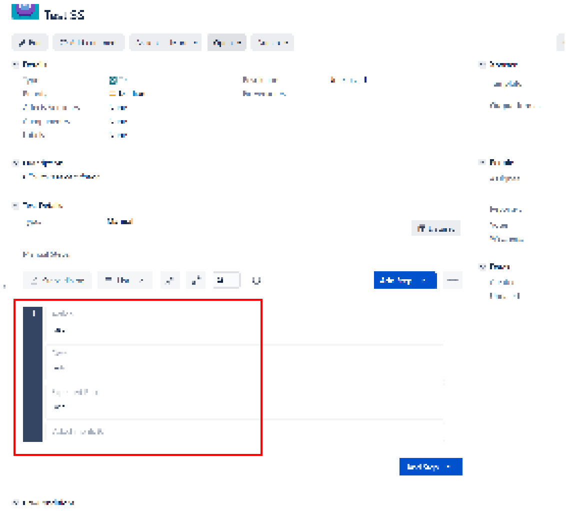


Fields

External Field		Xray Field
TCID (e.g. 1)	→	Test Case Identifier
Test Summary (e.g. steps)	→	Summary
Step (e.g. asd)	→	Step
Data (e.g. ads)	→	Data
Result (e.g. asd)	→	Expected Result

Press Next, and the import will start.

The ending result will be this (again with the example issue):



Link for related documentation

Rest API authentication

Insomnia

Xray's GraphQL Schema Documentation