

# Exporting Test Execution

- [Exporting Tests from a Test Execution](#)
- [Exporting Test Plans from a Test Execution](#)
- [Exporting Test Runs from a Test Execution](#)
- [Test Run Custom Fields](#)
- [Exporting Parameterized Tests from a Test Run](#)

From a **Test Execution**, it's possible to export the following Xray data: **Tests**, **Test Runs** and **Test Plans**.

All fields listed [here](#) are available because the **Test Execution** is a regular JIRA issue. In addition, you can export the following fields:

## Some mappings we can export from a Test Execution

```
Test Environments: ${TestEnvironments}

#Overall Execution Status (percentage + total of testes per status)
TO DO: ${Overall Execution Status.TO DO.Percentage}% - ${Overall Execution Status.TO DO.Count}
EXECUTING: ${Overall Execution Status.EXECUTING.Percentage}% - ${Overall Execution Status.EXECUTING.Count}
PASSED: ${Overall Execution Status.PASSED.Percentage}% - ${Overall Execution Status.PASSED.Count}
FAILED: ${Overall Execution Status.FAILED.Percentage}% - ${Overall Execution Status.FAILED.Count}
ABORTED: ${Overall Execution Status.ABORTED.Percentage}% - ${Overall Execution Status.ABORTED.Count}
```

## Exporting Tests from a Test Execution

Below you can find an example of how to iterate over the list of **Tests** associated with a **Test Execution**.

### Iterate over tests

```
Total number of associated tests: ${TestsCount}

// Iterating each test
#{for tests}
    Key: ${Tests[n].Key}
    Test Status: ${Tests[n].TestStatus}
    // Iterating over test steps for each test
    #{for m=Tests[n].TestStepsCount}
        Action: ${Tests[n].TestSteps[m].Action}
        Data: ${Tests[n].TestSteps[m].Data}
        Expected Result: ${Tests[n].TestSteps[m].ExpectedResult}
    #{end}
#{end}

or

// Iterating each test
#{for j=TestsCount}
    Key: ${Tests[j].Key}
    Test Status: ${Tests[j].TestStatus}
    // Iterating over test steps for each test
    #{for m=Tests[j].TestStepsCount}
        Action: ${Tests[j].TestSteps[m].Action}
        Data: ${Tests[j].TestSteps[m].Data}
        Expected Result: ${Tests[j].TestSteps[m].ExpectedResult}
    #{end}
#{end}
```



For more detailed information on how to export **Tests** and its information please check the [documentation](#).

## Exporting Test Plans from a Test Execution

Below you can find an example of how to iterate over the list of **TestPlans** associated with a **Test Execution**.


**Iterate over test plans**

```
Total number of associated test plans: ${TestPlansCount}

// Iterating each test plan
#{for testPlans}
    Summary: ${TestPlans[n].Summary}
#{end}

or

// Iterating each test plan
#{for j=TestPlansCount}
    Description: ${TestPlans[j].Description}
#{end}
```



For more detailed information on how to export **Test Plans** and its information please check the [documentation](#).

## Exporting Test Runs from a Test Execution


All fields listed [here](#) are available and will refer to the test execution associated. In addition, you can export the following fields:

Assigneeld	The test execution's assignee.
Comment	The test execution's comment.
Generic Test Definition	The generic test definition of the test run. <div>WIKI</div>
Cucumber Scenario	The cucumber definition of the test run
Executed By	The user's ID that executed the test execution.
Execution Status	The status of the test execution.
Started On	The started date of the test execution.
Finished On	The end date of the test execution.
ExecutionEvidencesCount	The number of execution evidences.
ExecutionDefectsCount	The number of execution defects.
TestStepsCount	The test execution's steps count.
TestType	The test execution's test type.
TestStepsDefectsCount	The number of defects for all test steps.

## Test Run Custom Fields

Xporter supports custom fields for Test Runs. To use your own defined custom fields, just use the same notation:

```
${<Custom field name>}
```



The custom field must have a different name from the table above.

**Iterate over test runs**

```
Total number of associated test runs: ${TestRunsCount}

// Iterating each test run
```

```

#{for testruns}
  Assignee Id: ${TestRuns[n].AssigneeId}
  Key: ${TestRuns[n].Key}
  Summary: ${TestRuns[n].Summary}
  Comment: ${TestRuns[n].Comment}
  Generic Test Definition: ${wiki:TestRuns[n].Generic Test Definition}
  Cucumber Scenario: ${TestRuns[n].Cucumber Scenario}
  Executed by: ${TestRuns[n].Executed By}
  Execution Status: ${TestRuns[n].Execution Status}
  Started on: ${TestRuns[n].Started On}
  Finished on: ${TestRuns[n].Finished On}
  Test Type: ${TestRuns[n].TestType}

  // Iterating over execution evidences for each test run
  Execution Evidences Total: ${TestRuns[n].ExecutionEvidencesCount}
  #{for m=TestRuns[n].ExecutionEvidencesCount}
    Id: ${TestRuns[n].ExecutionEvidences[m].Id}
    Name: ${TestRuns[n].ExecutionEvidences[m].Name}
    File URL: ${TestRuns[n].ExecutionEvidences[m].FileURL}
    Evidence Image: ${TestRuns[n].ExecutionEvidences[m].Evidence}
  #{end}

  // Iterating over execution defects for each test run
  Execution Defects Total: ${TestRuns[n].ExecutionDefectsCount}
  #{for m=TestRuns[n].ExecutionDefectsCount}
    Id: ${TestRuns[n].ExecutionDefects[m].Id}
    Description: ${TestRuns[n].ExecutionDefects[m].Description}
    Key: ${TestRuns[n].ExecutionDefects[m].Key}
    Summary: ${TestRuns[n].ExecutionDefects[m].Summary}
  #{end}

  // Iterating over test steps for each test run
  Test Steps Total: ${TestRuns[n].TestStepsCount}
  #{for m=TestRuns[n].TestStepsCount}
    Action: ${wiki:TestRuns[n].TestSteps[m].Action}
    Comment: ${TestRuns[n].TestSteps[m].Comment}
    Data: ${wiki:TestRuns[n].TestSteps[m].Data}
    Expected Result: ${wiki:TestRuns[n].TestSteps[m].ExpectedResult}
    Actual Result: ${wiki:TestRuns[n].TestSteps[m].ActualResult}
    Status: ${TestRuns[n].TestSteps[m].Status}
    Step Number: ${TestRuns[n].TestSteps[m].StepNumber}

    // Iterating over attachments for each test step
    Test Steps Attachments Count: ${TestRuns[n].TestSteps[m].AttachmentsCount}
    #{for l=TestRuns[n].TestSteps[m].AttachmentsCount}
      Id: ${TestRuns[n].TestSteps[m].Attachments[l].Id}
      Name: ${TestRuns[n].TestSteps[m].Attachments[l].Name}
      Attachment Image: ${TestRuns[n].TestSteps[m].Attachments[l].Attachment}
    #{end}

    // Iterating over evidences for each test step
    Test Steps Evidences Count: ${TestRuns[n].TestSteps[m].EvidencesCount}
    #{for l=TestRuns[n].TestSteps[m].EvidencesCount}
      Id: ${TestRuns[n].TestSteps[m].Evidences[l].Id}
      Name: ${TestRuns[n].TestSteps[m].Evidences[l].Name}
      File URL: ${TestRuns[n].TestSteps[m].Evidences[l].FileURL}
      Image: ${TestRuns[n].TestSteps[m].Evidences[l].Evidence}
    #{end}

    // Iterating over defects for each test step
    Test Steps Evidences Count: ${TestRuns[n].TestSteps[m].DefectsCount}
    #{for l=TestRuns[n].TestSteps[m].DefectsCount}
      Description: ${TestRuns[n].TestSteps[m].Defects[l].Description}
      Id: ${TestRuns[n].TestSteps[m].Defects[l].Id}
      Key: ${TestRuns[n].TestSteps[m].Defects[l].Key}
      Summary: ${TestRuns[n].TestSteps[m].Defects[l].Summary}
    #{end}
  #{end}
#{end}

```

The word document below shows an example with all the mappings available to export from a **TestExecution**:



## Exporting Parameterized Tests from a Test Run

To export all this data, we first need to execute the test run. If we change any data, we need to return to the Execution Page and merge it with the new data.

Fields	Description
ParametersCount	Total of Parameters declared in Dataset
IsDataDriven	Returns "true" if the current test run is data-driven; otherwise, it returns "false"
IterationsOverallExecutionStatus	List of all the statuses in the current test run iteration and their percentages
IterationsOverallExecutionStatus.STATUS.Count	The number of STATUS in the test run iterations
Iterations Overall Execution Status.STATUS.Percentage	The percentage of STATUS in the test run iterations
Parameters	List of the parameters from each test run iteration



### Data-driven

The field IsDataDriven is going to return TRUE only if the test run has more than one iteration.

## Exporting Test Runs Parameters from a Test Execution

For each Test Run Parameter you can export the following fields:

- Key
- Value

Below you can find an example of how to iterate over the list of **Test Run Parameters** associated with a **Test Execution**.

### Some mappings we can export from Test Runs Parameters

```
// Iterating each test run
#{for testruns}
  // Iterating over parameters for each test run
  Parameters Total: ${TestRuns[n].ParametersCount}
  #{for m=TestRuns[n].ParametersCount}
    Key: ${TestRuns[n].Parameters[m].Key}
    Value: ${TestRuns[n].Parameters[m].Value}
  #{end}
#{end}
```

## Exporting Test Runs Preconditions from a Test Execution

Below you can find an example of how to iterate over the list of **Test Run Preconditions** associated with a **Test Execution**.

### Some mappings we can export from Test Runs Preconditions

```
// Iterating each test run
#{for testruns}
  // Iterating over Preconditions for each test run
  #{for l=TestRuns[n].PreConditionsCount}
    Definition: ${TestRuns[n].PreConditions[l].PreCondition.Definition}
    Type: ${TestRuns[n].PreConditions[l].PreCondition.Type}
  #{end}
#{end}
```

## Exporting Test Runs Iterations from a Test Execution

For each Test Run Iteration you can export the following fields:

- Overall Execution Status
- Parameters from Test Run Iterations
- Test steps from Test Run Iterations

Below you can find an example of how to iterate over the list of **Test Run Iterations** associated with a **Test Execution**.

### Some mappings we can export from Test Run Iterations

```
// Iterating each test run
#{for testruns}
    IsDataDriven: ${TestRuns[n].IsDataDriven}

    // Iterations Overall Execution Status (percentage + total of testes per status)
    List of Statuses: ${TestRuns[n].IterationsOverallExecutionStatus}
    TO DO: ${TestRuns[n].IterationsOverallExecutionStatus.TO DO.Percentage}% - ${TestRuns[n].
IterationsOverallExecutionStatus.TO DO.Count}
    EXECUTING: ${TestRuns[n].IterationsOverallExecutionStatus.EXECUTING.Percentage}% - ${TestRuns[n].
IterationsOverallExecutionStatus.EXECUTING.Count}
    PASSED: ${TestRuns[n].IterationsOverallExecutionStatus.PASSED.Percentage}% - ${TestRuns[n].
IterationsOverallExecutionStatus.PASSED.Count}
    FAILED: ${TestRuns[n].IterationsOverallExecutionStatus.FAILED.Percentage}% - ${TestRuns[n].
IterationsOverallExecutionStatus.FAILED.Count}
    ABORTED: ${TestRuns[n].IterationsOverallExecutionStatus.ABORTED.Percentage}% - ${TestRuns[n].
IterationsOverallExecutionStatus.ABORTED.Count}

    // Iterating over test runs iterations
    Total of Iterations from a Test Run: ${TestRuns[n].IterationsCount}
    #{for m=TestRuns[n].IterationsCount}
        Name: ${TestRuns[n].Iterations[m].Name}
        Status: ${TestRuns[n].Iterations[m].Status}
        Parameters: ${TestRuns[n].Iterations[m].Parameters}

        // Iterating over parameters for each test run iteration
        Parameters Total: ${TestRuns[n].Iterations[m].ParametersCount}
        #{for l=TestRuns[n].Iterations[m].ParametersCount}
            Key: ${TestRuns[n].Iterations[m].Parameters[l].Key}
            Value: ${TestRuns[n].Iterations[m].Parameters[l].Value}
        #{end}

        // Iterating over preconditions for each test run iteration
        Preconditions Total: ${TestRuns[n].Iterations[m].PreConditionsCount}
        #{for l=TestRuns[n].Iterations[m].PreConditionsCount}
            Definition: ${TestRuns[n].Iterations[m].PreConditions[l].PreCondition.Definition}
            Type: ${TestRuns[n].Iterations[m].PreConditions[l].PreCondition.Type}
        #{end}

        // Iterating over test steps for each test run iteration
        #{for i=TestRuns[n].Iterations[m].TestStepsCount}
            Step Number: ${TestRuns[n].Iterations[m].TestSteps[i].StepNumber}
            Action: ${TestRuns[n].Iterations[m].TestSteps[i].Action}
            Data: ${TestRuns[n].Iterations[m].TestSteps[i].Data}
            Expected Result: ${TestRuns[n].Iterations[m].TestSteps[i].ExpectedResult}
            Status: ${TestRuns[n].Iterations[m].TestSteps[i].Status}
            Comment: ${TestRuns[n].Iterations[m].TestSteps[i].Comment}
            Actual Result: ${TestRuns[n].Iterations[m].TestSteps[i].ActualResult}

            // Iteration Test Step Attachments
            #{for l=TestRuns[n].Iterations[m].TestSteps[i].AttachmentsCount}
                Id: ${TestRuns[n].Iterations[m].TestSteps[i].Attachments[l].Id}
                Name: ${TestRuns[n].Iterations[m].TestSteps[i].Attachments[l].Name}
                Image: ${TestRuns[n].Iterations[m].TestSteps[i].Attachments[l].Attachment}
                FileURL: ${TestRuns[n].Iterations[m].TestSteps[i].Attachments[l].FileURL}
            #{end}
        #{end}
    #{end}
```

```

// Iteration Test Step Evidences
#{for l=TestRuns[n].Iterations[m].TestSteps[i].EvidencesCount}
    Id: ${TestRuns[n].Iterations[m].TestSteps[i].Evidences[l].Id}
    Name: ${TestRuns[n].Iterations[m].TestSteps[i].Evidences[l].Name}
    File URL: ${TestRuns[n].Iterations[m].TestSteps[i].Evidences[l].FileURL}
    Evidence: ${TestRuns[n].Iterations[m].TestSteps[i].Evidences[l].Evidence}
#{end}
// Iteration Test Step Defects
#{for l=TestRuns[n].Iterations[m].TestSteps[i].DefectsCount}
    Description: ${TestRuns[n].Iterations[m].TestSteps[i].Defects[l].Description}
    Id: ${TestRuns[n].Iterations[m].TestSteps[i].Defects[l].Id}
    Key: ${TestRuns[n].Iterations[m].TestSteps[i].Defects[l].Key}
    Summary: ${TestRuns[n].Iterations[m].TestSteps[i].Defects[l].Summary}
#{end}
#{end}
#{end}
#{end}

```

The word document below shows an example with all the mappings available to export from a **TestExecution**:

[blocked URLXray\\_Param\\_Test\\_Executions.docx](#)



**wiki**

indicates that the field supports the wiki format. More about [here](#).