

# Xray Test Management for Jira

Actual Result

 Available on **Xporter 3.3.0** or later and **Xray for Jira 1.8.0**.

- [Exporting a Test](#)
  - [Manual Test Steps](#)
  - [Manual Test Steps Attachments](#)
  - [Pre-Conditions associated with a Test](#)
  - [Requirements associated with a Test](#)
  - [Test Plans associated with a Test](#)
  - [Test Runs associated with a Test](#)
- [Exporting a Test Set](#)
- [The Xporter for Jira allows rendering all Tests of a Test Set.](#)
- [Exporting a Test Plan](#)
- [Exporting a Test Execution](#)
  - [Iterating Test Runs associated with a Test Execution](#)
  - [Iterating Pre-Conditions associated with a Test Run](#)
  - [Iterating Attachments associated with a Test Run](#)
  - [Iterating Evidences associated with a Test Run](#)
  - [Iterating Defects associated with a Test Run](#)
  - [Iterating Test Steps Attachments associated with a Test Run](#)
  - [Iterating Automated Test details associated with a Test Run](#)
  - [Iterating Manual Test Step details associated with a Test Run](#)
  - [Iterating Manual Test Step Attachments associated with a Test Run](#)
  - [Iterating Manual Test Step Defects associated with a Test Run](#)
  - [Iterating Manual Test Step Evidences associated with a Test Run](#)
  - [Test Executions associated with a Test Plan](#)
  - [Requirement Status](#)
  - [Xporter 5.8.0 improvements](#)
- [Compatibility Matrix](#)

## Exporting a Test

 On Xporter 5.4.0, if you have attachments enabled (see [Exporting Attachments](#) documentation), the Test Step Attachments are added to the generated document attachments.

## Manual Test Steps

Xporter for Jira allows rendering all Manual Test Steps of a Test.

**Definition:**

```
§ {TestSteps[n].Field}
```

*n* is the index of Test Step, starting from 0. The field **TestStepsCount** was created in order to give the total number of Test Steps. The fields available for Test Steps are:

- StepNumber
- Step
- Data
- ExpectedResult

**Expand to see the example on sample code**

```
#{for teststeps}
  §{TestSteps[n].StepNumber}
  §{TestSteps[n].Step}
  §{TestSteps[n].Data}
  §{TestSteps[n].ExpectedResult}
#{end}
```

You can also directly reference the first and last Test Step of a Test.

**Definition:**

```
    ${TestSteps[First].Field}  
    ${TestSteps[Last].Field}
```

**Expand to see the example on sample code**

```
    ${TestSteps[First].StepNumber}  
    ${TestSteps[First].Step}  
    ${TestSteps[Last].StepNumber}  
    ${TestSteps[Last].Step}
```

You can also Iterate over all Manual Test Steps while iterating over all Tests of a Test Set.

**Expand to see the example on sample code**

```
#{for tests}  
    ${Tests[n].Key}  
    ${Tests[n].Summary}  
    #{for m=Tests[n].TestStepsCount}  
        ${Tests[n].TestSteps[m].Step}  
        ${Tests[n].TestSteps[m].Data}  
        ${Tests[n].TestSteps[m].ExpectedResult}  
    #{end}  
#{end}
```

## Manual Test Steps Attachments

Xporter for Jira allows rendering all Attachments of Manual Test Steps of a Test. **This feature is available on Xporter 5.4.0 and Xray 3.0.0 and above**

**Definition:**

```
    ${TestSteps[n].Attachments[m].Field}
```

n is the index of Test Step and m is the index of the Attachment. The field **AttachmentsCount** was created in order to give the total number of Attachments. The fields available for Test Step Attachments are:

- ID
- Name
- Author
- AuthorFullName
- Created
- Size
- HumanReadableSize
- MimeType
- FileURL

**Expand to see the example on sample code**

```
#{for teststeps}  
    #{for m=TestSteps[n].AttachmentsCount}  
        ${TestSteps[n].Attachments[m].ID}  
        ${TestSteps[n].Attachments[m].Name}  
        ${TestSteps[n].Attachments[m].Author}  
        ${TestSteps[n].Attachments[m].AuthorFullName}  
        ${TestSteps[n].Attachments[m].Created}  
        ${TestSteps[n].Attachments[m].Size}  
        ${TestSteps[n].Attachments[m].HumanReadableSize}  
        ${TestSteps[n].Attachments[m].MimeType}  
        ${TestSteps[n].Attachments[m].FileURL}  
    #{end}  
#{end}
```

You can also directly reference the first and last Test Step Attachment of a Test.

**Definition:**

```
`${TestSteps[n].Attachments[First].Field}`  
`${TestSteps[n].Attachments[Last].Field}`
```

**Expand to see the example on sample code**

```
`${TestSteps[n].Attachments[First].ID}`  
`${TestSteps[n].Attachments[First].Name}`  
`${TestSteps[n].Attachments[First].Author}`  
`${TestSteps[n].Attachments[First].AuthorFullName}`  
`${TestSteps[n].Attachments[First].Created}`  
`${TestSteps[n].Attachments[Last].Size}`  
`${TestSteps[n].Attachments[Last].HumanReadableSize}`  
`${TestSteps[n].Attachments[Last].MimeType}`  
`${TestSteps[n].Attachments[Last].FileURL}`
```

You can also iterate over all Manual Test Steps Attachments while iterating over all Tests of a Test Set.

**Expand to see the example on sample code**

```
#{for tests}  
  `${Tests[n].Key}`  
  `${Tests[n].Summary}`  
  #{for m=Tests[n].TestStepsCount}  
    `${Tests[n].TestSteps[m].Step}`  
    `${Tests[n].TestSteps[m].Data}`  
    `${Tests[n].TestSteps[m].ExpectedResult}`  
    #{for i=Tests[n].TestSteps[m].AttachmentsCount}  
      `${Tests[n].TestSteps[m].Attachments[i].ID}`  
      `${Tests[n].TestSteps[m].Attachments[i].Name}`  
      `${Tests[n].TestSteps[m].Attachments[i].Author}`  
    #{end}  
  #{end}  
#{end}
```

## Pre-Conditions associated with a Test

Xporter for Jira allows rendering of all the Pre-Conditions associated with a Test.

**Definition:**

```
`${PreConditions[n].Field}`
```

n is the index of the Pre-Condition, starting from 0. The field **PreConditionsCount** was created in order to give the total number of Pre-Conditions.

Since Pre-Condition is a Jira Issue, you can render all the normal mappings which you are used to.

Example:

#### Expand to see the example on sample code

```
#{for preconditions}
  ${PreConditions[n].Key}
  ${PreConditions[n].Summary}
  ${PreConditions[n].Description}
  ${PreConditions[n].Conditions}
#{end}

or

#{for j=PreConditionsCount}
  ${PreConditions[j].Key}
  ${PreConditions[j].Summary}
  ${PreConditions[j].Description}
  ${PreConditions[j].Conditions}
#{end}
```

You can also Iterate over all the Pre-Conditions associated with a Test using an [Xray Enhanced querying JQL Function](#):

#### Expand to see the example on sample code

```
#{for i=JQLIssuesCount|clause=key in testPreConditions('${Key}')}
  ${JQLIssues[i].Key}
  ${JQLIssues[i].Summary}
  ${JQLIssues[i].Description}
  ${JQLIssues[i].Conditions}
#{end}
```

You can also Iterate over all the Pre-Conditions associated with a Test while iterating over all Tests of a Test Set.

#### Expand to see the example on sample code

```
#{for tests}
  ${Tests[n].Key}
  ${Tests[n].Summary}
  #{for m=Tests[n].PreConditionsCount}
    ${Tests[n].PreConditions[m].Key}
    ${Tests[n].PreConditions[m].Summary}
    ${Tests[n].PreConditions[m].Description}
    ${Tests[n].PreConditions[m].Conditions}
  #{end}
#{end}
```

## Requirements associated with a Test

Xporter for Jira allows rendering of all the Requirements associated with a Test using an [Xray Enhanced querying JQL Function](#), where you input the given Test Key:

**JQL Function:**

```
testRequirements('${Key}')
```

In order to give the total of Requirements associated with a given Test, you can use the following JQL Count statement, where you input the given Test Key:

```
#{jqlcount:testRequirements('${Key}')
```

n is the index of the Pre-Condition, starting from 0. The field **PreConditionsCount** was created in order to give the total number of Pre-Conditions.

Since a Requirement is a Jira Issue, you can render all the normal mappings which you are used to.

Example:

#### Expand to see the example on sample code

```
#{for k=JQLIssuesCount|clause=key in testRequirements('${Key}')}
  ${JQLIssues[k].Key}
  ${JQLIssues[k].Summary}
  ${JQLIssues[k].Description}
#{end}
```

You can also iterate over all the Requirements associated with a Test while iterating over all Tests of a Test Set.

#### Expand to see the example on sample code

```
#{for tests}
  ${Tests[n].Key}
  ${Tests[n].Summary}
  #{for k=JQLIssuesCount|clause=key in testRequirements('Tests[n].Key')}
    ${Tests[n].JQLIssues[k].Key}
    ${Tests[n].JQLIssues[k].Summary}
    ${Tests[n].JQLIssues[k].Description}
  #{end}
#{end}
```

## Test Plans associated with a Test

Xporter for Jira allows rendering of all the Test Plans associated with a Test.

### Definition:

```
#{TestPlans[n].Field}
```

n is the index of the Test Plan, starting from 0. The field **TestPlansCount** was created in order to give the total number of Test Plans.

Since a Test Plan is a Jira Issue, you can render all the normal mappings which you are used to.

Example:

#### Expand to see the example on sample code

```
#{for testPlans}
  ${TestPlans[n].Key}
  ${TestPlans[n].Summary}
  ${TestPlans[n].Description}
#{end}
```

You can also directly reference the first and last Test Plan of a Test.

#### Definition:

```
`${TestPlans[First].Field}`
`${TestPlans[Last].Field}`
```

You can also iterate over all the Test Plans associated with a Test while iterating over all Tests of a Test Set.

#### Expand to see the example on sample code

```
#{for tests}
  ${Tests[n].Key}
  ${Tests[n].Summary}
  #{for m=Tests[n].TestPlansCount}
    ${Tests[n].TestPlans[m].Key}
    ${Tests[n].TestPlans[m].Summary}
    ${Tests[n].TestPlans[m].Description}
  #{end}
#{end}
```

## Test Runs associated with a Test

Xporter for Jira allows rendering of all the Test Runs associated with a Test.

#### Definition:

```
`${TestRuns[n].Field}`
```

$n$  is the index of the Test Run, starting from 0. The field **TestRunsCount** was created in order to give the total number of Test Runs.

Example:

#### Expand to see the example on sample code

```
##Iterating over Test Runs
#{for testruns}
  Execution Status: ${TestRuns[n].Execution Status}
  AssigneeID: ${TestRuns[n].AssigneeId}
  Rank: ${TestRuns[n].Rank}
  Executed By: ${TestRuns[n].Executed By}
  Started On: ${TestRuns[n].Started On}
  Finished On: ${TestRuns[n].Finished On}
  Comment: ${wiki:TestRuns[n].Comment}
  Execution Defects Count: ${TestRuns[n].ExecutionDefectsCount}
  TestSteps Defects Count: ${TestRuns[n].TestStepsDefectsCount}
  Evidences Count: ${TestRuns[n].ExecutionEvidencesCount}
#{end}
```

Furthermore, you can iterate over the test run and get all the information from the test run directly from the test. You just need to use the same syntax as you have [here](#) when we iterate over the test runs from test execution.

# Exporting a Test Set

The Xporter for Jira allows rendering all Tests of a Test Set.

As it is not known in advance how many Tests exist in a Test Set, there is a notation that allows iterating a section over all the Tests of a Test Set. The notation is:

### Expand to see the examples on sample code

```
#Test Set Status (percentage)
Todo: ${Test Set Status.TODO.Percentage} %
Fail: ${Test Set Status.FAIL.Percentage} %
Pass: ${Test Set Status.PASS.Percentage} %
Executing: ${Test Set Status.EXECUTING.Percentage} %
Aborted: ${Test Set Status.ABORTED.Percentage} %

#Test Set Status Percentage + total of testes per status
Todo: ${Test Set Status.TODO.Percentage} % (${Test Set Status.TODO.Count})
Fail: ${Test Set Status.FAIL.Percentage} % (${Test Set Status.FAIL.Count})
Pass: ${Test Set Status.PASS.Percentage} % (${Test Set Status.PASS.Count})
Executing: ${Test Set Status.EXECUTING.Percentage} % (${Test Set Status.EXECUTING.Count})
Aborted: ${Test Set Status.ABORTED.Percentage} % (${Test Set Status.ABORTED.Count})

#{for tests}
    Content and Test Mappings.          Example: ${Tests[n].Key}
#{end}

or

#{for j=TestsCount}
    Content and Test Mappings.          Example: ${Tests[j].Summary}
#{end}
```

n is the index of Test, starting from 0. The field **TestsCount** was created in order to give the total of Tests.

You can also directly reference the first and last Test of a Test Set.

### Definition:

```
`${Tests[First].Field}`
`${Tests[Last].Field}`
```

### Expand to see the examples on sample code

```
`${Tests[First].Key}`
`${Tests[First].Summary}`
`${Tests[Last].Key}`
`${Tests[Last].Summary}`
```

# Exporting a Test Plan

The Xporter for Jira allows rendering all Tests of a Test Plan.

As it is not known in advance how many Tests exist in a Test Plan, there is a notation that allows iterating a section over all the Tests of a Test Plan. The notation is:

### Expand to see the examples on sample code

```
#Overall Execution Status (percentage + total of testes per status)
Todo: ${Overall Execution Status.TODO.Percentage}% (Overall Execution Status.TODO.Count)
Fail: ${Overall Execution Status.FAIL.Percentage}% (Overall Execution Status.FAIL.Count)
Pass: ${Overall Execution Status.PASS.Percentage}% (Overall Execution Status.PASS.Count)
Executing: ${Overall Execution Status.EXECUTING.Percentage}% (Overall Execution Status.EXECUTING.Count)
Aborted: ${Overall Execution Status.ABORTED.Percentage}% (Overall Execution Status.ABORTED.Count)

#{for tests}
    Content and Test Mappings.          Example: ${Tests[n].Key}
#{end}

or

#{for j=TestsCount}
    Content and Test Mappings.          Example: ${Tests[j].Summary}
#{end}
```

n is the index of Test, starting from 0. The field **TestsCount** was created in order to give the total number of Tests.

You can also directly reference the first and last Test of a Test Plan.

#### Definition:

```
${Tests[First].Field}
${Tests[Last].Field}
```

### Expand to see the examples on sample code

```
${Tests[First].Key}
${Tests[First].Summary}
${Tests[Last].Key}
${Tests[Last].Summary}
```

## Exporting a Test Execution



On Xporter 5.4.0, if you have attachments enabled (see [Exporting Attachments](#) documentation), the Test Execution Evidences, the Test Step Evidences and the Test Step Attachments are added to the generated document attachments.

Xporter for Jira allows rendering Test Execution Issue details. The notation is:

### Expand to see the examples on sample code of a Test Execution details

```
#Overall Execution Status (percentage + total of testes per status)
Todo: ${Overall Execution Status.TODO.Percentage}% (Overall Execution Status.TODO.Count)
Fail: ${Overall Execution Status.FAIL.Percentage}% (Overall Execution Status.FAIL.Count)
Pass: ${Overall Execution Status.PASS.Percentage}% (Overall Execution Status.PASS.Count)
Executing: ${Overall Execution Status.EXECUTING.Percentage}% (Overall Execution Status.EXECUTING.Count)
Aborted: ${Overall Execution Status.ABORTED.Percentage}% (Overall Execution Status.ABORTED.Count)
Custom Test Run Status: ${Blocked}

#Total of Test Runs
Test Runs Count: ${TestRunsCount}
```



If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Test Runs associated with a Test Execution

Xporter for Jira allows rendering of all the Test Runs associated with a Test Execution.

**Definition:**

```
${TestRuns[n].Field}
```

n is the index of the Test Run, starting from 0. The field **TestRunsCount** was created in order to give the total number of Test Runs.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
    Execution Status: ${TestRuns[n].Execution Status}
    AssigneeID: ${AssigneeId}
    Rank: ${TestRuns[n].Rank}
    Executed By: ${TestRuns[n].Executed By}
    Started On: ${TestRuns[n].Started On}
    Finished On: ${TestRuns[n].Finished On}
    Comment: ${wiki:TestRuns[n].Comment}
    Execution Defects Count: ${TestRuns[n].ExecutionDefectsCount}
    TestSteps Defects Count: ${TestRuns[n].TestStepsDefectsCount}
    Evidences Count: ${TestRuns[n].ExecutionEvidencesCount}
#{end}
```



If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Pre-Conditions associated with a Test Run

Xporter for Jira allows rendering of the Pre-Conditions associated with a Test from a Test Run.

**Definition:**

```
${TestRuns[n].PreCondition.Field}
```

n is the index of Test Runs, starting from 0. The fields available for Pre-Conditions are:

- Key
- Summary
- Conditions
- Pre-Condition Type

### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
    #Getting data from Pre-Condition
    Pre-Condition Key: ${TestRuns[n].PreCondition.Key}
    Pre-Condition Summary: ${TestRuns[n].PreCondition.Summary}
    Condition:          ${TestRuns[n].PreCondition.Conditions}
    Type: ${TestRuns[n].PreCondition.Pre-Condition Type}
#{end}
```

 The fields defined [here](#) can be obtained from the fields Pre-Condition and Defects.

 If a Test Execution contains a lot of information it can decrease Jira performance

## Iterating Attachments associated with a Test Run

Xporter for Jira allows rendering of all the Attachments associated with a Test Run.

**Definition:**

```
${TestRuns[n].AttachmentsCount[sa]}
```

sa is the index of the Attachments, starting from 0. The field **AttachmentsCount** was created in order to give the total number of Attachments of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

**Expand to see the example on sample code**

```
#Iterating over Test Runs
#{for testruns}
  #Iterating Test Attachments
  #{for sa=TestRuns[n].AttachmentsCount}
    Name: ${TestRuns[n].Attachments[sa].Name}
    Author: ${TestRuns[n].Attachments[sa].Author}
    ID: ${TestRuns[n].Attachments[sa].ID}
    Size: ${TestRuns[n].Attachments[sa].Size}
  #{end}
#{end}
```

 If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Evidences associated with a Test Run

Xporter for Jira allows rendering of all the Evidences associated with a Test Run.

**Definition:**

```
${TestRuns[n].ExecutionEvidences[d]}
```

d is the index of the Evidences, starting from 0. The field **ExecutionEvidencesCount** was created in order to give the total number of Evidences of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over Evidences
  #{for d=TestRuns[n].ExecutionEvidencesCount}
    Id: ${TestRuns[n].ExecutionEvidences[d].Id}
    Name: ${TestRuns[n].ExecutionEvidences[d].Name}
    Author: ${TestRuns[n].ExecutionEvidences[d].Author}
    Link: @{{title=${TestRuns[n].ExecutionEvidences[d].FileURL}|href=${TestRuns[n].ExecutionEvidences
[d].FileURL}}
    Size: ${TestRuns[n].ExecutionEvidences[d].Size}
    Created: ${TestRuns[n].ExecutionEvidences[d].Created}
    HumanReadableSize: ${TestRuns[n].ExecutionEvidences[d].HumanReadableSize}
    MimeType: ${TestRuns[n].ExecutionEvidences[d].MimeType}
    Evidence:${TestRuns[n].ExecutionEvidences[d].Evidence}
  #{end}
#{end}
```



The field Evidence will print the image if the execution evidence on the test run is an image



If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Defects associated with a Test Run

Xporter for Jira allows rendering of all the defects associated with a Test Run.

### Definition:

```
${TestRuns[n].ExecutionDefects[e]}
```

e is the index of the defects, starting from 0. The field **ExecutionDefectsCount** was created in order to give the total number of Defects of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over defects from each test run
  #{for e=TestRuns[n].ExecutionDefectsCount}
    Link: @{{title=${TestRuns[n].ExecutionDefects[e].Key}|href=${BaseUrl}/browse/${TestRuns[n].
ExecutionDefects[e].Key}}
    Summary: ${TestRuns[n].ExecutionDefects[e].Summary}
  #{end}
#{end}
```



If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Test Steps Attachments associated with a Test Run

Xporter for Jira allows rendering of all the Test Steps Attachments associated with a Test Run.

### Definition:

```
${TestRuns[n].TestStepsAttachments[a]}
```

a is the index of the Test Steps Attachments, starting from 0. The field **TestStepsAttachmentsCount** was created in order to give the total of Test Steps Attachments of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

**Expand to see the example on sample code**

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over TestSteps Attachments
  #{for a=TestRuns[n].TestStepsAttachmentsCount}
    Name: ${TestRuns[n].TestStepsAttachments[a].Name}
    Author: ${TestRuns[n].TestStepsAttachments[a].Author}
    Link: @{{title=${TestRuns[n].TestStepsAttachments[a].FileURL}|href=${TestRuns[n].
TestStepsAttachments[a].FileURL}}
    Size: ${TestRuns[n].TestStepsAttachments[a].Size}
  #{end}
#{end}
```

 If a Test Execution contains lots of information it can decrease Jira performance

## Iterating Automated Test details associated with a Test Run

Xporter for Jira allows rendering of the Details from Automated Tests associated with a Test Run.

**Definition:**

**Cucumber Scenario:** `${TestRuns[n].Cucumber Scenario}`

**Test Definition:** `${TestRuns[n].Generic Test Definition}`

n is the index of the Test Runs, starting from 0. The fields **Cucumber Scenario/Generic Test Definition** were created in order to give the step details of Automated Tests of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

**Expand to see the example on sample code**

```
#Iterating over Test Runs
#{for testruns}
  #Test Run Step Details
  Cucumber Scenario: ${TestRuns[n].Cucumber Scenario}
  Test Definition: ${TestRuns[n].Generic Test Definition}
#{end}
```

 If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Manual Test Step details associated with a Test Run

Xporter for Jira allows rendering of the Details from Manual Tests associated with a Test Run.

**Definition:**

`${TestRuns[n].TestSteps[r]}`

r is the index of the Test Steps, starting from 0. The field **TestStepsCount** was created in order to give the step details of Manual Tests of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

**Expand to see the example on sample code**

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over Manual Test Steps from Test Run
  #{for r=TestRuns[n].TestStepsCount}
    StepNumber:      ${TestRuns[n].TestSteps[r].StepNumber}
    Step: ${wiki:TestRuns[n].TestSteps[r].Step}
    Data: ${wiki:TestRuns[n].TestSteps[r].Data}
    Expected Result: ${wiki:TestRuns[n].TestSteps[r].ExpectedResult}
    Comment: ${wiki:TestRuns[n].TestSteps[r].Comment}
    Status:      ${TestRuns[n].TestSteps[r].Status}
    Actual Result: ${wiki:TestRuns[n].TestSteps[r].Actual Result}
  #{end}
#{end}
```

 If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Manual Test Step Attachments associated with a Test Run

Xporter for Jira allows rendering of the Attachments from Manual Tests Steps associated with a Test Run.

**Definition:**

```
${TestRuns[n].TestSteps[r].Attachments[sa]}
```

sa is the index of the Test Step Attachments, starting from 0. The field **AttachmentsCount** was created in order to give the step attachments of Manual Tests of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

**Expand to see the example on sample code**

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over Manual Test Steps from Test Run
  #{for r=TestRuns[n].TestStepsCount}
    #Iterating over Test Step Attachments
    #{for sa=TestRuns[n].TestSteps[r].AttachmentsCount}
      Name: {TestRuns[n].TestSteps[r].Attachments[sa].Name}
      Author: {TestRuns[n].TestSteps[r].Attachments[sa].Author}
      Link: {title={TestRuns[n].TestSteps[r].Attachments[sa].FileURL}|href={TestRuns[n].
TestSteps[r].Attachments[sa].FileURL}}
      Size: {TestRuns[n].TestSteps[r].Attachments[sa].Size}
    #{end}
  #{end}
#{end}
```

 If a Test Execution contains a lot of information, it can decrease Jira performance

## Iterating Manual Test Step Defects associated with a Test Run

Xporter for Jira allows rendering of the Defects from Manual Tests Steps associated with a Test Run.

**Definition:**

```
${TestRuns[n].TestSteps[r].Defects[dc]}
```

dc is the index of the Test Step Defects, starting from 0. The field **DefectsCount** was created in order to give the step defects of Manual Tests of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

#### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over Manual Test Steps from Test Run
  #{for r=TestRuns[n].TestStepsCount}
    #Iterating over Test Step Defects
    #{for dc=TestRuns[n].TestSteps[r].DefectsCount}
      Link: {title={TestRuns[n].TestSteps[r].Defects[dc].Key}|href={BaseURL}/browse/{TestRuns
[n].TestSteps[r].Defects[dc].Key}}
      Summary: {wiki:TestRuns[n].TestSteps[r].Defects[dc].Summary}
    #{end}
  #{end}
#{end}
```



If a Test Execution contains a lot of information, it can decrease Jira performance



The fields defined [here](#) can be obtained for fields Pre-Condition and Defects.

## Iterating Manual Test Step Evidences associated with a Test Run

Xporter for Jira allows rendering of the Evidences from Manual Tests Steps associated with a Test Run.

**Definition:**

```
${TestRuns[n].TestSteps[r].Evidences[e]}
```

e is the index of the Test Step Evidences, starting from 0. The field **EvidencesCount** was created in order to give the step evidences of Manual Tests of a Test Run.

Since a Test Run isn't a Jira Issue, you can render only the following mappings.

Example:

#### Expand to see the example on sample code

```
#Iterating over Test Runs
#{for testruns}
  #Iterating over Manual Test Steps from Test Run
  #{for r=TestRuns[n].TestStepsCount}
    #Iterating over Test Step Evidences
    #{for e=TestRuns[n].TestSteps[r].EvidencesCount}
      Name: ${TestRuns[n].TestSteps[r].Evidences[e].Name}
      Author: ${TestRuns[n].TestSteps[r].Evidences[e].Author}
      Link: @{{title=${TestRuns[n].TestSteps[r].Evidences[e].FileURL}|href=${TestRuns[n].
TestSteps[r].Evidences[e].FileURL}}
      Size: ${TestRuns[n].TestSteps[r].Evidences[e].Size}
      Created: ${TestRuns[n].TestSteps[r].Evidences[e].Created}
      HumanReadableSize: ${TestRuns[n].TestSteps[r].Evidences[e].HumanReadableSize}
      MimeType: ${TestRuns[n].TestSteps[r].Evidences[e].MimeType}
      Evidence: ${TestRuns[n].TestSteps[r].Evidences[e].Evidence}
    #{end}
  #{end}
#{end}
```



If you want to export the images, for example `${TestRuns[n].ExecutionEvidences[d].FileURL}` you can check [here](#) for instructions on how to do it.



If a Test Execution contains a lot of information, it can decrease Jira performance.

## Test Executions associated with a Test Plan

Xporter for Jira allows rendering of all the Test Executions that belong to a Test Plan.

### Definition:

```
${TestExecution[n].Field}
```

`n` is the index of the Test Plan, starting from 0. The field **TestExecutionsCount** was created in order to give the total number of Test Executions.

Since a Test Execution is a Jira Issue, you can render all the normal mappings that you are used to.

You can also extract all the information that you normally do regarding a Test Execution.

Example:

### Expand to see the example on sample code

```
#{for testExecutions}
  ${TestExecutions[n].Key}
  ${TestExecutions[n].Summary}
  ${TestExecutions[n].Description}
  #{for i=TestExecutions[n].TestRunsCount}
    ${TestExecutions[n].TestRuns[i].Executed By}
    ${TestExecutions[n].TestRuns[i].PreCondition.Key}
    .....
  #{end}
#{end}
```

You can see all the info that can be extracted for a Test Execution in the section above.

You can also directly reference the first and last Test Execution of a Test Plan.

### Definition:

```
${TestExecutions[First].Field}
${TestExecutions[Last].Field}
```

## Xporter 5.8.0 improvements

With Xporter 5.8.0+ combined with Xray 3.3.0+, we made some major performance improvements.

With these versions, it is now possible to access Test Execution information by iterating over tests and its Test Runs.

**Expand to see the example on sample code**

```
#{for tests}
  ${Tests[n].Key}
  #{for c=Tests[n].TestRunsCount}
    ${Tests[n].TestRuns[c].Executed By}
    ${Tests[n].TestRuns[c].PreCondition.Key}

    ${Tests[n].TestRuns[c].TestExecution.Key}
    ${Tests[n].TestRuns[c].TestExecution.Summary}
    ${Tests[n].TestRuns[c].TestExecution.Description}
    .....
  #{end}
#{end}
```

## Compatibility Matrix

Xporter for Jira Version	Xray Version
3.3.0, 3.3.1, 3.3.2	1.8.0 - 1.9.0
3.4.0 - 4.3.5	1.9.1 - 1.12.x
4.3.7	1.9.1+
5.0.X	2.0.1 - 2.1.2
5.1	2.0.1 - 2.2.0
5.4.0 - 5.6.5	3.0.0 - 3.1.5
5.7.0 - 5.75	3.1.7 - 3.2.3
5.8.0 - 6.2.9	3.3.0-3.3.6
6.3.0+	3.4.0+

**Expand to see the example on sample code**

```
#{for i=JQLIssuesCount|clause=
  ${PreConditions[n].Key}
  ${PreConditions[n].Summary}
  ${PreConditions[n].Description}
#{end}

or

#{for j=PreConditionsCount}
  ${PreConditions[j].Key}
  ${PreConditions[j].Summary}
  ${PreConditions[n].Description}
#{end}
```