Exporting a Test Execution

Document Generator allows you to get following data from the Xray Test Execution:

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

Overall Execution Status

Export the Overall Execution Status with name and percentage for each Test Execution Status

```
${Overall Execution Status}
```

You can print the status of the Test Execution by using the following notation:

% per Status	Number of Tests per Status
${Overall Execution Status.NameOfStatus}	${Overall Execution Status.NameOfStatus.Count}

See the real example:

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

```
Todo: ${Overall Execution Status.TODO}% (${Overall Execution Status.TODO.Count})
Fail: ${Overall Execution Status.FAIL}% (${Overall Execution Status.FAIL.Count})
Pass: ${Overall Execution Status.PASS}% (${Overall Execution Status.PASS.Count})
Executing: ${Overall Execution Status.EXECUTING}% (${Overall Execution Status.EXECUTING.Count})
Aborted: ${Overall Execution Status.ABORTED}% (${Overall Execution Status.ABORTED.Count})
```

Iterating Test Runs associated with a Test Execution

Document Generator allows rendering of all the Test Tuns 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}
```

Exporting Test Runs custom fields

To export Test Runs Custom Fields you just have to defined the placeholder with the name of you custom field.

Example: Image that you have a custom field called "Run CF". To get the value printed on you document you just have to use the following placeholder:

```
#{for testruns}
The Run CF value is: ${TestRuns[n].Run CF}
#{end}
```

> If your custom field type is a Number, Data or Date Time you can use formatting functions.

Iterating Pre-Conditions associated with a Test Run

Document Generator 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}
```

Iterating Attachments associated with a Test Run

Document Generator 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

Document Generator 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}
```

Iterating Defects associated with a Test Run

Document Generator 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}
```

Iterating Automated Test details associated with a Test Run

Document Generator 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

Document Generator 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}
                Action: ${wiki:TestRuns[n].TestSteps[r].Action}
                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}
```

Iterating Manual Test Step Attachments associated with a Test Run

Document Generator 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

Document Generator 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}
```

Iterating Manual Test Step Evidences associated with a Test Run

Document Generator 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.

Exporting Test Run Activity

Document Generator allows export all the activity of a Test Run.

Definition:

$ {TestRuns[n].ActivityEntries[ac]}

ac is the index of the Activity entry, starting from 0. The field **ActivityEntriesCount** was created in order to give the Activity entry of a Test Run.

Example:

Expand to see the example on sample code

```
#{for testruns}
#{for d=TestRuns[n].ActivityEntriesCount}
Action: ${TestRuns[n].ActivityEntries[d].Action}
Author: ${TestRuns[n].ActivityEntries[d].Author}
Created at: ${dateformat("dd-MM-yyyy HH:mm:ss"):TestRuns[n].ActivityEntries[d].Created}
Changes:
#{for ch=TestRuns[n].ActivityEntries[d].ActivityItemsCount}
Field: ${TestRuns[n].ActivityEntries[d].ActivityItems[ch].Field}
OldValue: ${TestRuns[n].ActivityEntries[d].ChangedItems[ch].OldValue}
NewValue: ${TestRuns[n].ActivityEntries[d].ChangedItems[ch].NewValue}

#{end}
#{end}
#{end}
```