

# JQL

- [Inserting the number of issues returned by a JQL Query](#)
- [FilterJQL & FilterName](#)
- [Filter](#)
- [JQL limit of issues](#)

## Inserting the number of issues returned by a JQL Query

Xporter for Jira allows you to insert in the document the number of issues returned by the execution of a JQL query.

**Definition:**

```
${jqlcount:<JQL Query>}
```

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

```
${jqlcount:Project = DEMO}
```

The example above will put the number of issues of the project DEMO in the document.

## FilterJQL & FilterName



This only works when the export is made via the **Export** menu on the Issue Navigator screen.

Xporter for Jira also allows you to render the FilterJQL and the FilterName used to search Issues on the Issue Navigator screen.

**Definition:**

```
${FilterJQL}
```

```
${FilterName}
```

## Filter

You can now work with the filters you have defined on your Jira instance.  
If you want to get the JQL statement, simply map:

```
${filter:<Filter name> or <Filter ID>}
```

Furthermore, if you want to export the jqlCount, you don't need to put the entire JQL Query, you just need use the filter name or the filter ID to see all the entries you have on this filter:

```
${jqlcount:${filter:<Filter name> or <Filter ID>}}
```

Last but not least, you can iterate issues with the JQL associated to a filter within the clause parameter by passing only the filter name or filter ID:

**Expand to see the sample code**

A simple example iterating the details of issues from a filter already defined on your Jira

```
#{for i=JQLIssuesCount|clause=${filter:<Filter Name> or <FilterId>}}
  ${JQLIssues[i].Key}
  ${JQLIssues[i].Summary}
#{end}
```



When the filter is shared, we must use <Filter Id> instead of <Filter Name>, so that the same template is used by all members who have access to the filter.

## JQL limit of issues

Once the user defines a maximum number of issues on Global Settings, JQL issues processing will replace the iteration that exceeds that limit with a warning message.

This improvement will avoid the JIRA performance decreasing in cases which the number of issues to be export is a large number. This validation will be done in all kinds of exports provided by Xporter.

The maximum number of issues is 10:

### Bulk Export Options

Break Pages

This option controls if and when page breaks are made while exporting multiple issues.

Maximum number of issues

This option controls how many issues is a user allowed to export at one time, using a multiple issues export.

Maximum number of simultaneous requests

This option controls how many xporter processes can be run simultaneously.

The first filter returns fewer issues than the defined limit and the second returns more than 10 issues:

First:

Key: DEMO-9

IssueType: Story

Priority: Medium

Summary: As a developer, I'd like to update story status during the sprint >> Click the Active sprints link at the top right of the screen to go to the Active sprints where the current Sprint's items can be updated

**Xporter could not process this block because it exceeds the maximum number of allowed issues. Issues to process 12, limit defined 10. Used JQL Query: key <= DEMO-12 order by Key desc.**

First:

Key: DEMO-8

IssueType: Bug

Priority: Medium

Summary: As a product owner, I'd like to include bugs, tasks and other issue types in my backlog >> Bugs like this one will also appear in your backlog but they are not normally estimated

**Xporter could not process this block because it exceeds the maximum number of allowed issues. Issues to process 12, limit defined 10. Used JQL Query: key <= DEMO-12 order by Key desc.**

[illegible]