

Raley IntakeForms Filters - JIFF

JIFF, JIF Filters or Raley Intake Forms Filters is the core concept of Raley Intake Forms. It helps you to define the rules and conditions for your fields and make forms more dynamic and "smart".

Syntax for JIF Filters is similar to what you can see in Atlassian CQL or in Apache Lucene.

Wildcards are supported everywhere, as well as logical "AND" and "OR" operators.

JIF Filters operate on internal field names and operate on field values (which means that for dropdown fields, JIF Filters operate on "IDs", not the labels)

Field name for Raley Intake Form is visible in the configuration

The screenshot shows the configuration interface for a field named "priority". At the top, there is a header bar with a menu icon, the field name "Priority", a dropdown menu, and a trash icon. Below this, the configuration area is divided into sections: "Name" with the value "priority" and a description "Name of the field to be used when writing conditions / filters"; "Label" with the value "Priority" and a description "Label is shown next the the field control"; "Required" with an unchecked checkbox and a description "When 'required' attribute is set, field label will have the ** sign appended"; and "Description" with an empty text area and a description "Description is shown under the field on the form".

Let's take this "priority" field as an example

It is a standard JIRA field for setting the issue priority.

When rendered on the form it renders as a dropdown field like this

The screenshot shows the "Priority" dropdown field as it appears on a form. It consists of the label "Priority" followed by a dropdown menu.

And actually has 5 values by default

ID	Label
1	Highest
2	High
3	Medium
4	Low
5	Lowest

In HTML the controls looks like this:

```
<select data-field="priority" title="Priority" type="select" class="select" name="priority" id="i_priority">
  <option value=""></option>
  <option value="1">Highest</option>
  <option value="2">High</option>
  <option value="3">Medium</option>
  <option value="4">Low</option>
  <option value="5">Lowest</option>
</select>
```

For example, we want to define the rule which will show a field when "priority" field is selected as "Highest"

The condition will be written like this then

priority:1

And the rule might look like this

+ Add Field Rule

Field Rule

Will be executed when the field value changes

Action Show field ▼

Condition priority:1

When this rules should be executed.
[Jirassimo Intake Forms Filter \(JIFF\)](#) is expected here,
 leave blank or put *, if the rule has to be executed
 everytime

Field name description|

Name of the field

OK
Close

Which will be translated to - when a priority field has the highest priority selected show a field with name "description" (for example, this field is set to be hidden initially, on the form load, by using the "form rules")

More examples on JIF Filters

JIF Filter	Details
*	Matches all, same as when the condition is left blank

summary:*urgent*	When summary field contains the word (full or partly) that match to "urgent"
!summary:*urgent*	You can negate the filter with ! sign
priority:[empty] AND summary:*urgent*	When priority is not specified and the summary field has the word "urgent"
priority:[empty] !priority:[empty]	Checking for empty and non empty
summary:*risk* OR label:highrisk	When summary field contains the word "risk" and the label field is set to "highrisk"
(summary:[empty] AND priority:1) OR (!summary:[empty] AND priority:2)	You can have multiple conditions wrapped in brackets

You can also use the following functions with fields

Function	What it does	Example
_length	Returns the length of the field. For empty or non-existent fields the length is 0	Checks if summary field's length is longer than 20 symbols summary._length:>20
_matches(<i>Regular expression</i>)	You can validate the field value against the given regular expression. Returns true or false	Checks if value is not empty myfield._matches:^(?=\s*\S).* myfield._matches:^(?=\s*\S).*: true Value ends with pdf myfield._matches(.*?\pdf) myfield._matches(.*?\pdf):true Value DOES NOT end with pdf myfield._matches(.*?\pdf) myfield._matches(.*?\pdf):false