

# Migrate 'Condition based on regular expression'

The [Condition based on regular expression of JWT DC](#) is not yet available in JWT Cloud but can be implemented easily using a [Jira expression condition](#) .

Since JWT DC field codes are not available in conditions or validators in JWT Cloud, they have to be replaced by their Jira expression equivalent.

## Migration steps

1

Add a [Jira expression condition](#).

2



Build a matching Jira expression by migrating the parameters of the [Condition based on regular expression](#).

The resulting Jira expression has the form

```
VALUE.match(Regular_expression) != null
```

**VALUE** and **Regular\_expression** are the transformed parameters of [Condition based on regular expression](#) which are migrated like shown in the following table.

## Migration details

 JWT DC option JWT DC	 JWT Cloud	Notes						
Value	Value from field	Select the corresponding field from the Field injector.						
	Set value manually (parser expression)	Translate the JWT DC parser expression to a <a href="#">Jira expression</a> .						
Regular expression		Enter the regular expression of the JWT DC parameter.						
Additional options	Case insensitive	Prefix the regular expression by ( ?i ) without white space between ( ?i ) and the remaining regular expression.  Example: <table><tr><th>JWT DC regular expression</th><th>Jira expression regular expression</th></tr><tr><td>Important Task</td><td>( ?i )Important Task</td></tr><tr><td>.*[JWT]</td><td>( ?i ).*[JWT]</td></tr></table>	JWT DC regular expression	Jira expression regular expression	Important Task	( ?i )Important Task	.*[JWT]	( ?i ).*[JWT]
	JWT DC regular expression	Jira expression regular expression						
Important Task	( ?i )Important Task							
.*[JWT]	( ?i ).*[JWT]							
	Multi-line	Prefix the regular expression by ( ?m ) without white space between ( ?m ) and the remaining regular expression. In case you already have a ( ? ) prefix, add m there.						

	Dot all	Replace the "." by "(. \n)" in the regular expression.	
	Literal	Escape each of the special characters of the regular expression, e.g. <div>\.\\*\[JWT\\]</div>	
	Negate condition	Replace the != operator by ==	

Due to the different architecture, it may happen that the condition gets too complex. This is the case when many fields are checked. The condition cannot be saved, and a corresponding error message will be displayed. If that's the case, the condition has to be split up into two or more.

If you still have questions, feel free to refer to our [support](#) team.