

Prevent the creation of sub-tasks unless the parent issue is in a certain status

On this page

[Configuration](#) | [Related use cases](#)

Use case

In this use case, we will prevent the creation of sub-tasks when the parent issue of the sub-tasks is in any status different from the one that we specify.



Configuration

1

Add the [Logical validator](#) to the **Create** workflow transition.

2

Expression

Add the following expression replacing the name of the status with the status of your choice:

```
%{parent.key} != null IMPLIES %{parent.status} =~ "In Progress"
```

With this solution the user will get an error message when he clicks on *Create* after filling up the creation screen. A solution that prevents the user from filling up the sub-task's creation screen is not possible.

3

Error message

Add an error message as the following one.

Sub-tasks can only be created when the parent issue is in the status In progress.



Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Block an Epic's transition depending on linked issues status and due date	 	Logical validator Logical condition	count() filterByPredicate() linkedIssues()	

Block an Epic's transition until all the issues under that Epic are resolved		Logical validator Logical condition	count() issuesUnderEpic() filterByResolution()	
Block a transition until all sub-tasks have certain fields populated	 	Logical validator Logical condition	count() filterByPredicate() subtasks()	STAFF PICK
Ensure that all issues linked with a certain issue link type have "Due Date" field set	 	Logical validator Logical condition	count() linkedIssues() fieldValue()	
Ensure that an issue has at least one attachment		Logical validator Logical condition		STAFF PICK
Evaluate Assets objects	 	Logical validator Logical condition	findPattern() findReplaceAll() replaceAll() toStringList()	
Halt a transition if an element is not contained in a list	 	Logical validator Logical condition	toStringList()	
Make "Time spent" field required		Logical validator		
Prevent a closed issue from being reopened after resting 1 week closed	 	Logical validator Logical condition	isInRole()	
Prevent a transition depending of the number of components	 	Logical validator Logical condition	numberOfSelectedItems()	
Prevent external users from creating issues		Logical validator	matches() toString() textOnStringList() toStringList() userDisplayName()	
Prevent issue creation if description contains less than 100 characters		Logical validator	replaceAll() length()	
Prevent issue creation with the same field value (Boolean)		Logical validator Validation based on JQL query	count() issuesFromJQL()	
Prevent setting due dates outside business hours		Logical validator Logical condition	withinCalendar()	STAFF PICK
Prevent the creation of sub-tasks unless the parent issue is in a certain status		Logical validator		

Reject duplicated file names in attachments	 	Logical validator	<code>count()</code>
		Logical condition	<code>toStringList()</code> <code>distinct()</code>
Restrict issue creation per issue type and project role		Logical validator	<code>isInRole()</code>
Validate a Select List (cascading) custom field		Logical validator Logical condition	
Validate only issue links created in transition screen		Logical validator	<code>count()</code>
			<code>transitionLinkedIssues()</code>
			<code>filterByProject()</code>
			<code>filterByIssueType()</code>
Validation based on the value of a date type project property		Logical validator	<code>stringToDate()</code>
		Logical condition	<code>projectProperty()</code>

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