

Prevent a closed issue from being reopened after resting 1 week closed

On this page

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

Use case

Prevent a certain issue from being reopened after seven days closed, unless user has the role of "**Administrator**" or "**Supervisor**".



Configuration

1

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

2

Expression

Add the following expression:

```
%{issue.status} = "Closed" AND {system.currentDateTime} - {issue.lastStatusChange} > 7 * {DAY} IMPLIES isInRole(%{system.currentUser}, "Administrator") OR isInRole(%{system.currentUser}, "Supervisor")
```



Screenshots

Update parameters of the 'Logical validator' Validator for this transition.

Update parameters of the 'Logical validator' Validator for this transition.

Basic configuration*

Define the required parameters for this validator.

Expression*

```
1 %{issue.status} = "Closed" AND {system.currentDateTime} - {issue.lastStatusChange} > 7 * {DAY} IMPLIES isInRole(%{system.currentUser}, "Administrator") OR isInRole(%{system.currentUser}, "Supervisor")
```

The expression must return **true** or **false**. The condition is met if the **logical expression** returns **true**.

› [Validation options](#)

› [Error message](#)

Logical validator

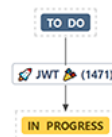
[Documentation](#)



Evaluate a logical expression that returns true or false - a most universal and powerful function to compare field values across any kind of issue relation, count sub-tasks or links.




















Workflow












JWT 3.0.0 Use cases



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()	
Hide transition to a previous status		Logical condition		
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	<code>replaceAll() length()</code>	
Prevent issue creation with the same field value (Boolean)		Logical validator Validation based on JQL query	<code>count()</code> <code>issuesFromJQL()</code>	
Prevent setting due dates outside business hours		Logical validator Logical condition	<code>withinCalendar()</code>	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 Logical condition	<code>count()</code> <code>toStringList()</code> <code>distinct()</code>	
Restrict issue creation per issue type and project role		Logical validator	<code>isInRole()</code>	
Set a condition in a global transition which only applies in a certain status		Logical condition		
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 Logical condition	<code>stringToDate()</code> <code>projectProperty()</code>	

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