

Ensure that all issues linked with a certain issue link type have "Due Date" field set

On this page

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

Use case

Check that the issue linked to the current issue with link type name "linkA" has a **due date** field set.



Configuration

1

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

2

Expression

Add the following expression:

```
count(linkedIssues("linkA")) = count(fieldValue({issue.dueDate}, linkedIssues("linkA")))
```



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 count(linkedIssues("linkA")) = count(fieldValue({issue.dueDate}, linkedIssues("linkA")))
```

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

› [Validation options](#)

› [Error message](#)

Logical validator

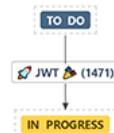
[Documentation](#)



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	count()	
		Logical condition	filterByPredicate() linkedIssues()	
Block an Epic's transition until all the issues under that Epic are resolved		Logical validator	count()	
		Logical condition	issuesUnderEpic() filterByResolution()	
Block a transition until all sub-tasks have certain fields populated	 	Logical validator	count()	STAFF PICK
		Logical condition	filterByPredicate() subtasks()	
Ensure that all issues linked with a certain issue link type have "Due Date" field set	 	Logical validator	count()	
		Logical condition	linkedIssues() fieldValue()	
Ensure that an issue has at least one attachment	 	Logical validator		STAFF PICK
		Logical condition		
Evaluate Assets objects	 	Logical validator	findPattern() findReplaceAll() replaceAll() toStringList()	
		Logical condition		
Halt a transition if an element is not contained in a list	 	Logical validator	toStringList()	
		Logical condition		
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	isInRole()	
		Logical condition		
Prevent a transition depending of the number of components	 	Logical validator	numberOfSelectedItems()	
		Logical condition		
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 Logical condition	count() toStringList() distinct()	
Restrict issue creation per issue type and project role		Logical validator	isInRole()	
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	count() transitionLinkedIssues() filterByProject() filterByIssueType()	
Validation based on the value of a date type project property		Logical validator Logical condition	stringToDate() projectProperty()	

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