

# Check project property

This function has been **renamed** with the [JWT 3.0](#) release.

Find the new documentation at:

[Condition on JWT project property](#)

[Validation of JWT project property](#)

## On this page

- [Purpose](#)
- [Example: Allowing issue reopening in certain projects, while forbidding it in other ones](#)
- [Usage Examples](#)
- [Related Features](#)

## Purpose

There is a **condition** and a **validator** available for checking the value of a [Project Properties](#). These features allows to **hide a transition** (condition) or **block the execution of a transition** if certain [Project Properties](#) doesn't contain an expected value.

These features are of great value for building **reusable workflows**, that behave somehow differently according to certain characteristics of the projects. Each particular characteristic of the project is set by means of a [Project Properties](#).

## Example: Allowing issue reopening in certain projects, while forbidding it in other ones

We are going to implement a validation for preventing issues to be reopened in projects where [Project Properties](#) **"allowReopening"** has value different from **"Yes"** or is missing.

We use [Check project property](#) validator in transition **"Reopen Issue"** using the following configuration:

<b>Project property name:</b>	ReopeningAllowed Field code injector: Summary - [Text] - %{00000} You can insert field codes (with format %{nnnnn}) that will be replaced into the string with the field values. Project properties are stored in project description with the following format: {property_name=property_value} e.g. {MaxNumberBugs=5}
<b>Comparison operator:</b>	<input checked="" type="radio"/> is equal <input type="radio"/> isn't equal <input type="radio"/> contains <input type="radio"/> doesn't contain <input type="radio"/> starts with <input type="radio"/> doesn't start with <input type="radio"/> ends with <input type="radio"/> doesn't end with
<b>String value to be compared to:</b>	Yes Field code injector: Summary - [Text] - %{00000} You can insert field codes into the string (with format %{nnnnn}) that will be replaced with the field values.
<b>Skip validation when:</b> Inhibit the validator under selected circumstances.	<input type="checkbox"/> Transition is triggered by a <a href="#">bulk operation</a> . <input type="checkbox"/> Transition is triggered by a <a href="#">JIRA Workflow Toolbox post-function</a> . <input type="checkbox"/> Current issue is being created by cloning. Only makes sense in <i>Create Issue</i> transition.

<b>Message to show when validation fails:</b>	Issue reopening is not allowed in current project. Field code injector: Summary - [Text] - %{00000} Field codes with format %{nnnnn} can be inserted in the failure message and its translations, and they will be replaced with actual field values at runtime. <a href="#">Hide translations for installed languages</a>
<b>Language</b>	<b>Validation Failure Message Translations</b>
alemán (Alemania)	
español (España)	No está permitida la reapertura de incidencias en el proyecto actual.
francés (Francia)	
inglés (Estados Unidos)	
inglés (UK)	
japonés (Japón)	

Once configured, transition "**Reopen Issue**" looks like this:

RESOLVED

CLOSED

Reopen Issue

REOPENED

Screen: [Workflow Screen](#)

Conditions 1

Validators 1

Post Functions 6

The transition requires the following criteria to be valid

Add validator

Project property **ReopeningAllowed** must be equal to "Yes".

Message to show when validation fails: **"Issue reopening is not allowed in current project."**, providing translations for the following languages: **español (España)**.

The project property is created in **project's description** as shown:

Edit Project: Customer Relationship Management

Name \*

Customer Relationship Management


Key \*

CRM

edit key

URL

Project Avatar \*



select image

Description

{ReopeningAllowed=Yes}

[Jira Workflow Toolbox](#) provides a [Expression Parser](#), which is used in the main features of the plugin. This parser provides a set of functions for checking the existence of a project property, and reading its value:

FUNCTION	RETURNED VALUE
----------	----------------

<b>projectProperty</b> (string <b>property_name</b> ) : string	Returns a string with the value of project property with name <b>property_name</b> in current issue's project. Shows an error message if project property doesn't exist. Example: <b>projectProperty("maxNumberOfReopenings")</b> returns "3", provided there is a string <b>{maxNumberOfReopenings=3}</b> in the description of current issue's project.
<b>projectProperty</b> (string <b>property_name</b> , string <b>project_key</b> ) : string Available since version 2.2	Returns a string with the value of project property with name <b>property_name</b> in project with key <b>project_key</b> . Shows an error message if project property doesn't exist. Example: <b>projectProperty("maxNumberOfReopenings", "CRM")</b> returns "3", provided there is a string <b>{maxNumberOfReopenings=3}</b> in the description of project with key "CRM".
<b>projectPropertyExists</b> (string <b>property_name</b> ) : boolean Available since version 2.2	Returns true only if there is a project property with name <b>property_name</b> in current issue's project, i.e., if project's description contains a string like <b>{property_name=value}</b> . Example: <b>projectPropertyExists("maxNumberOfReopenings")</b> returns true only if there is a string like <b>{maxNumberOfReopenings=x}</b> in the description of current issue's project.
<b>projectPropertyExists</b> (string <b>property_name</b> , string <b>project_key</b> ) : boolean Available since version 2.2	Returns true only if there is a project property with name <b>property_name</b> in project with key <b>project_key</b> . Example: <b>projectPropertyExists("maxNumberOfReopenings", "CRM")</b> returns true only if there is a string like <b>{maxNumberOfReopenings=x}</b> in the description of project with key "CRM".

The parser is a common feature, making [Project Properties](#) available in the main features of the plugin:

- Condition and Validator [Boolean condition and validator with math, date-time or text-string terms](#)
- Post-functions [Set a field as a function of other fields](#), [Copy parsed text to a field](#) and [Mathematical and date-time expression calculator](#)

## Usage Examples

## Related Features

- [Read a project property](#)
- [Set or create project property](#)