

# Condition and validation based on regular expression

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

Find the new documentation at:

[Condition based on regular expression](#)

[Validation based on regular expression](#)

## On this page

- [Purpose](#)
- [Example: How to validate US State codes entered in a custom field](#)
- [Configuration Parameters](#)
- [Usage Examples](#)
- [Related Features](#)

## Purpose

**Condition based on regular expression** and **Validation based on regular expression** are two forms of a same feature for checking the **value of a field** or a **text composed by various custom fields** against a **regular expression**.

---

## Example: How to validate US State codes entered in a custom field

In this example we show a configuration for validating the US state code entered in a custom field called **"State"**:

Field or parsed text to be checked for matching with the regular expression:

Field

Estate - [Text Field (single line)]

Parsed text

1

Field code injector:

Summary - [Text] - %{00000}

Field codes with format %{nnnn} will be replaced with the corresponding field values at runtime.

Regular expression:

1

^( (AL) | (AK) | (AS) | (AZ) | (AR) | (CA) | (CO) | (CT) | (DE) | (DC) | (FM) | (FL) | (GA) | (GU) | (HI) | (ID) | (IL) | (IN) | (IA) | (KS) | (KY) | (LA) | (ME) | (MH) | (MD) | (MA) | (MI) | (MN) | (MS) | (MO) | (MT) | (NE) | (NV) | (NH) | (NJ) | (NM) | (NY) | (NC) | (ND) | (MP) | (OH) | (OK) | (OR) | (PW) | (PA) | (PR) | (RI) | (SC) | (SD) | (TN) | (TX) |

Field code injector:

Summary - [Text] - %{00000}

Field codes with format %{nnnn} can be inserted in the regular expression, and will be replaced with the corresponding field values at runtime.

Regular expression syntax

Regular expression parsing modes:

Case Insensitive:

Case-insensitive matching is done in a manner consistent with the Unicode Standard.

Multiline:

Expressions ^ and \$ match just after or just before, respectively, a line terminator or the end of the input sequence. By default these expressions only match at the beginning and the end of the entire input sequence.

Dot All:

Expression . matches any character, including a line terminator. By default this expression does not match line terminators.

Literal:

Input string is treated as a sequence of literal characters. Metacharacters or escape sequences in the input sequence will be given no special meaning. Case-insensitive mode retains its impact on matching when used in conjunction with this mode.

Negate condition:

When this option is checked, validation will be satisfied if regular expression is **NOT MATCHED** by selected field or entered parsed text.

Skip validation when:

Inhibit the validator under selected circumstances.

Transition is triggered by a bulk operation.

Transition is triggered by a JIRA Workflow Toolbox post-function.

Current issue is being created by cloning. Only makes sense in Create Issue transition.

Message to show when validation fails:

Invalid US state code.

Field code injector:

Summary - [Text] - %{00000}

Field codes with format %{nnnn} can be inserted in the failure message and its translations, and they will be replaced with actual field values at runtime.

Hide translations for installed languages

Language	Validation Failure Message Translations
alemán (Alemania)	
español (España)	Código de estado de los EEUU inválido.
francés (Francia)	
inglés (Estados Unidos)	
inglés (UK)	
japonés (Japón)	

Regular expression used is: ^((AL)|(AK)|(AS)|(AZ)|(AR)|(CA)|(CO)|(CT)|(DE)|(DC)|(FM)|(FL)|(GA)|(GU)|(HI)|(ID)|(IL)|(IN)|(IA)|(KS)|(KY)|(LA)|(ME)|(MH)|(MD)|(MA)|(MI)|(MN)|(MS)|(MO)|(MT)|(NE)|(NV)|(NH)|(NJ)|(NM)|(NY)|(NC)|(ND)|(MP)|(OH)|(OK)|(OR)|(PW)|(PA)|(PR)|(RI)|(SC)|(SD)|(TN)|(TX)|(UT)|(VT)|(VI)|(VA)|(WA)|(WV)|(WI)|(WY))\$

Once configured, the validation will look like this:

Conditions 2	Validators 1	Post Functions 10
--------------	--------------	-------------------

The transition requires the following criteria to be valid

Add validator

If field **Estate** matches the following parsed regular expression: `^ ( (AL) | (AK) | (AS) | (AZ) | (AR) | (CA) | (CO) | (CT) | (DE) | (DC) | (FM) | (FL) | (GA) | (GU) | (HI) | (ID) | (IL) | (IN) | (IA) | (KS) | (KY) | (LA) | (ME) | (MH) | (MD) | (MA) | (MI) | (MN) | (MS) | (MO) | (MT) | (NE) | (NV) | (NH) | (NJ) | (NM) | (NY) | (NC) | (ND) | (MP) | (OH) | (OK) | (OR) | (PW) | (PA) | (PR) | (RI) | (SC) | (SD) | (TN) | (TX) | (UT) | (VT) | (VI) | (VA) | (WA) | (WV) | (WI) | (WY) ) $`.

Message to show when validation fails: **"Invalid US state code."**, providing translations for the following languages: **español (España)**.

## Configuration Parameters

We will explain in detail each of the configuration parameters of these features.

### Field or parsed text to be checked for matching with the regular expression

These are two alternative parameters, i.e., only one of these parameters is used in each configuration at the same time.

- **Field:** Field whose value is going to be checked against the regular expression.
- **Parsed text:** Text composed by string literals and field codes. The text obtained once parsed, will be checked against the regular expression.

Field codes have the following formats:

- **%{nnnnn}**: for any field type. This field code will be replaced with the value of the field in current issue. Cascading Select List will be replaced with **parent\_value - child\_value**. Example: **%{00001}** is field code for issue **Description**.
- **%{nnnnn.i}**: for Cascading Select Lists fields. Index **i** can take value **0** for parent (or root) value, and **1** for child value. Example: **%{10100.0}** is field code for **parent value of a cascading select field with field ID 10100**.

**Multi-Level Cascading Select** custom fields are also supported. For these fields values higher 1 can be used for index **i**.

Field codes can be checked up in a dropdown list available in the configuration screen of these features.

### Regular expression

This is the regular expression that will be used for validating the value of the selected field or parsed text entered.

**JAVA regular expression syntax** is used.

Field codes with formats **%{nnnnn}** and **%{nnnnn.i}** can be inserted. Regular expression will be preprocessed before being evaluated, in order to replace each field code with its corresponding value. This way, dynamic regular expressions that depend on the value of fields in current issue, can be entered.

### Regular expression parsing modes

The way regular expressions are evaluated can be modulated with the following modes:

- **Case Insensitive:** Case-insensitive matching is done in a manner consistent with the Unicode Standard.
- **Multiline:** Expressions **^** and **\$** match just after or just before, respectively, a line terminator or the end of the input sequence. By default these expressions only match at the beginning and the end of the entire input sequence.
- **Dot All:** Expression **.** matches any character, including a line terminator. By default this expression does not match line terminators.
- **Literal:** Input string is treated as a sequence of literal characters. Metacharacters or escape sequences in the input sequence will be given no special meaning. Case-insensitive mode retains its impact on matching when used in conjunction with this mode.

### Negate condition

When this option is checked, **not matching** the regular expression is the requirement set for field value or parsed text, instead of matching it.

### Message to show when validation fails

This parameter is only available in the **validator** form of this feature. It's a **custom message** to explain the user the cause that is preventing him from executing the transition.

Translations\* to every language installed in the JIRA instance can be optionally entered.

In **Condition based on regular expression** the transition is simply hidden whenever condition isn't satisfied, so this parameter is not present.

---

## Usage Examples

Page: [Make "Affects Version/s" mandatory when issue resolution is "Fixed"](#)

Page: [Make a custom field mandatory when Priority is "Critical" or "Blocker"](#)

Page: [Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"](#)

Page: [Make certain custom field required in resolve screen only if the resolution was set to "Fixed"](#)

Page: [Prevent addition of new sub-tasks if the parent issue is in "Resolved" or "Closed" status](#)

Page: [Validation on issue attachments](#)

Page: [Validation on the value of a Cascading Select field](#)

## Related Features

- [Boolean condition and validator with math, date-time or text-string terms](#)