Validation on the value of a Cascading Select field

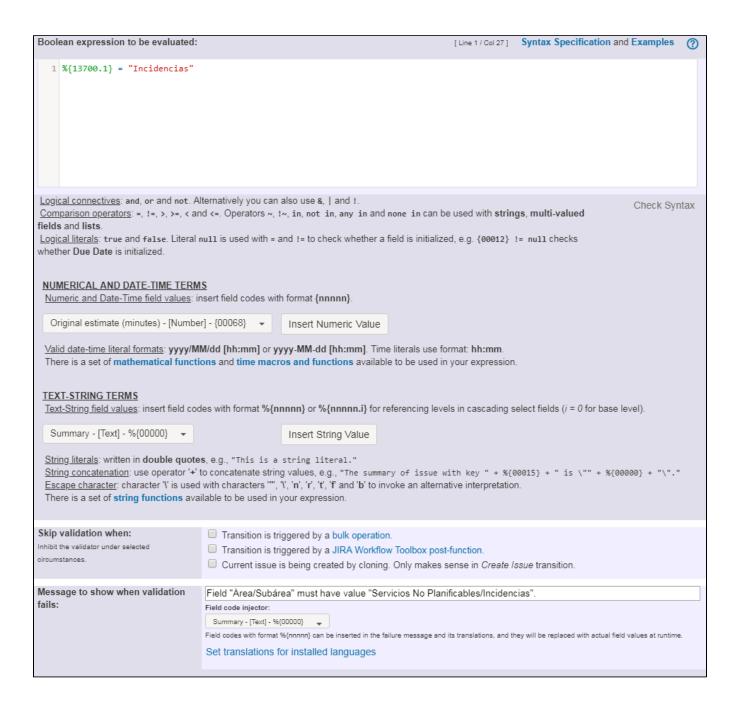## Features used to implement the example

- **Boolean validator with math, date-time or text-string terms**
- **Validation based on regular expression**

Example: Validation on the value of a Cascading Select field

1. I have an issue type **INCIDENCIA**
2. I have a custom field of Cascading Select type named "**Área/Subárea**".
3. The correct values for "**Área/Subárea**" area:
   a. Servicios No Planificables/Incidencias
   b. Servicios No Planificables/Servicios
   c. Servicios No Planificables/Cambios
   d. Proyectos/Proyectos menores
4. I need to validate that an issue of "**INCIDENCIA**" issue type cannot be created if "**Área/Subáreea**" is not equal to "**Servicios No Planificables/Incidencias**"
5. I have defined in **INCIDENCIAS** workflow

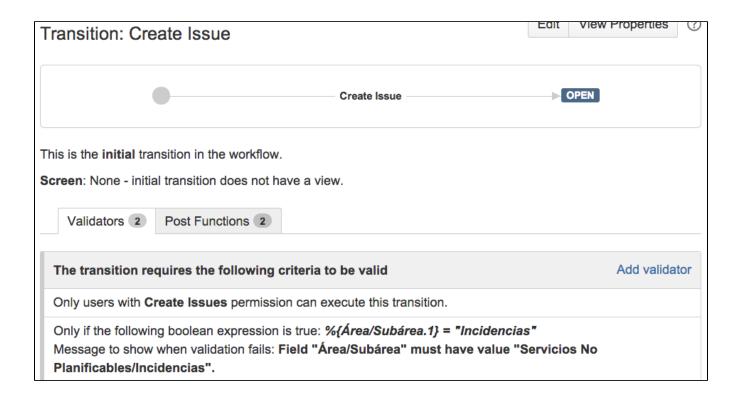If you use **Boolean validator with math, date-time or text-string terms** you have to use the following configuration:

```
1  %{13700.1} = "Incidencias"
```

Check Syntax

<u>Logical connectives</u>: `and`, `or` and `not`. Alternatively you can also use `&`, `|` and `!`.
<u>Comparison operators</u>: `=`, `!=`, `>`, `>=`, `<` and `<=`. Operators `~`, `!~`, `in`, `not in`, `any in` and `none in` can be used with **strings**, **multi-valued fields** and **lists**.
<u>Logical literals</u>: `true` and `false`. Literal `null` is used with `=` and `!=` to check whether a field is initialized, e.g. `{00012} != null` checks whether **Due Date** is initialized.

NUMERICAL AND DATE-TIME TERMS
<u>Numeric and Date-Time field values</u>: insert field codes with format **{nnnnn}**.

| Original estimate (minutes) - [Number] - {00068} ▾ | Insert Numeric Value |

<u>Valid date-time literal formats</u>: **yyyy/MM/dd [hh:mm]** or **yyyy-MM-dd [hh:mm]**. Time literals use format: **hh:mm**.
There is a set of **mathematical functions** and **time macros and functions** available to be used in your expression.

TEXT-STRING TERMS
<u>Text-String field values</u>: insert field codes with format **%{nnnnn}** or **%{nnnnn.i}** for referencing levels in cascading select fields ($i = 0$ for base level).

| Summary - [Text] - %{00000} ▾ | Insert String Value |

<u>String literals</u>: written in **double quotes**, e.g., `"This is a string literal."`
<u>String concatenation</u>: use operator '**+**' to concatenate string values, e.g., `"The summary of issue with key " + %{00015} + " is \"" + %{00000} + "\"."`
<u>Escape character</u>: character '\' is used with characters '"', '\', 'n', 'r', 't', 'f' and 'b' to invoke an alternative interpretation.
There is a set of **string functions** available to be used in your expression.

Skip validation when:	☐ Transition is triggered by a bulk operation.
Inhibit the validator under selected circumstances.	☐ 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:	Field "Àrea/Subárea" must have value "Servicios No Planificables/Incidencias".
	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.
	Set translations for installed languages

Text to be parsed: `%{13700.1} = "Incidencias"`

Note that:

Field code for field "**Área/Subárea**" is **%{13700}**.

This field code depends on each Jira instance. You can find the code for you particular Jira instance in the field list available the validator configuration screen.

Once configured, your transition will look like this:

Transition: Create Issue

Edit   View Properties   ?

Create Issue ────────────▶ OPEN

This is the **initial** transition in the workflow.

Screen: None - initial transition does not have a view.

| Validators **2** | Post Functions **2** |

The transition requires the following criteria to be valid   Add validator

Only users with **Create Issues** permission can execute this transition.

Only if the following boolean expression is true: *%{Área/Subárea.1} = "Incidencias"*
Message to show when validation fails: **Field "Área/Subárea" must have value "Servicios No Planificables/Incidencias".**

You can also use another feature of **Jira Workflow Toolbox**: **Validation based on regular expression**. If you use that feature you should use the following configuration:

Field or parsed text to be checked for matching with the regular expression:	○ Field Summary - [Text] ▾	?
	● Parsed text `1 %{13700.1}`	
	Field code injector:	
	Summary - [Text] - %{00000} ▾	
	Field codes with format %{nnnnn} will be replaced with the corresponding field values at runtime.	

Regular expression:

`1 Incidencias`

Field code injector:

Summary - [Text] - %{00000} ▾

Field codes with format %{nnnnn} 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:

Field "Àrea/Subárea" must have value "Servicios No Planificables/Incidencias".

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.

Set translations for installed languages

Other examples of that function

Boolean validator with math, date-time or text-string terms

Page: Block a transition until all sub-tasks have certains fields populated
Page: Block an epic's transition depending on linked issues status and due date
Page: Block or hide a transition for an issue depending on its issue links
Page: Block or unblock a transition after an issue rested a specific time in a status
Page: Block transition until all sub-tasks are in a specific status category
Page: Close parent issue when all sub-tasks are closed
Page: Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
Page: Ensure that all issues linked with a certain issue link type have "Due Date" field set
Page: If field A is populated then, field B must also be populated
Page: Limit issue creation per role and issue type
Page: Limit the number of hours a user can log per day
Page: Limit valid dates for work logs
Page: Make "Time Spent" field required when there is no time logged in the issue
Page: Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"
Page: Make attachment mandatory depending on the value of certain custom field

Related Usage Examples

- Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B
    - example
    - validator
    - custom-field
- Validation on the value of a Cascading Select field
    - example
    - validator
    - custom-field
- Make different fields mandatory depending on the value of a Select List custom field
    - example
    - validator
    - custom-field
- Validate compatible values selection among dependent custom fields
    - example
    - validator
    - custom-field
- Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"
    - example
    - validator
    - custom-field

Validation based on regular expression