

Validate compatible values selection among dependent custom fields

On this page

- [Features used to implement the example](#)
- [Example: Validate compatible values selection among dependent custom fields](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

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

Example: Validate compatible values selection among dependent custom fields

I have three pull-down

- PD1 = (element11, element12, element13)
- PD2 = (element21, element22, element23)
- PD3 = (element31, element32, element33)

Logic goes like

- If (PD1.element11) is selected display PD2 & PD3
- If (PD1.element12) is selected display PD2, disable PD3
- If (PD1.element13) show or disable PD2 & PD3

With [Jira Workflow Toolbox](#) you can implement a validation in a transition (also at "Create Issue" transition) indicating incompatible value selection for fields **PD1**, **PD2** and **PD3**. To do it you can use [Boolean validator with math, date-time or text-string terms](#) with the following configuration:

Boolean expression to be evaluated:

```
{%{11800} != "element12" OR {%{11801} != null AND {%{11802} = null} AND (%{11800} != "element13" OR {%{11801} = null AND {%{11802} != null})
```

Logical connectives: or, and and not. Alternatively you can also use |, & and !.

Comparison operators: =, !=, >, >=, < and <=.

Logical literals: 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

Numeric and Date-Time field values: insert field codes with format `{nnnnn}`.

Original estimate (minutes) - [Number] - {00068}

INSERT NUMERIC VALUE

Valid date-time literal formats: `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

Text-String field values: insert field codes with format `%{nnnnn}` or `%{nnnnn.i}` for referencing levels in cascading select fields (*i* = 0 for base level).

PD3 - [Select List (single choice)] - {%{11802}

INSERT STRING VALUE

String literals: written in **double quotes**, e.g., *"This is a string literal."*

String concatenation: use operator '+' to concatenate string values, e.g., *"The summary of issue with key " + {%{00015} + " is \"" + {%{00000} + "\"."*

Escape character: 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.

Message to show when validation fails:

Incompatible values selected for fields PD2 and PD3 according to the value selected in PD1.

Text to be parsed is: `{%{11800} != "element12" OR {%{11801} != null AND {%{11802} = null} AND (%{11800} != "element13" OR {%{11801} = null AND {%{11802} != null)`

Note that field codes in the particular instance of Jira used for this example are:

- `%{11800}` is field code for custom field "PD1"
- `%{11801}` is field code for custom field "PD2"
- `%{11802}` is field code for custom field "PD3"

Once configured, transition "Create Issue" 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 3

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: **{PD1} != "element12" OR {PD2} != null AND {PD3} = null) AND ({PD1} != "element13" OR {PD2} = null AND {PD3} != null)**

Message to show when validation fails: **Incompatible values selected for fields PD2 and PD3 according to the value selected in PD1.**

Other examples of that function

Page: [Block a transition until all sub-tasks have certain 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](#)

Page: [Make different fields mandatory depending on the value of a Select List custom field](#)

Page: [Make linked issues, sub-tasks and JQL selected issues progress through its workflows](#)

Page: [Make parent issue progress through its workflow](#)

Page: [Prevent issue creation if another issue with same field value already exists](#)

Page: [Reject duplicated file names in attachments](#)

Page: [Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field](#)

Page: [Require issue link when resolving as duplicate](#)

Page: [Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status](#)

Page: [Restrict sub-task type creation depending on parent issue status](#)

Page: [Restrict sub-task type creation depending on parent issue type](#)

Page: [Set a condition in a global transition which only applies in a certain status](#)

Related Usage Examples

- [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](#)
- [Validate a custom field "Story Points" has been given a value in Fibonacci sequence](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Make attachment mandatory depending on the value of certain custom field](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Enforce a field \(Select List\) to be set when another field \(Radio Button\) has a certain value \(works with any kind of field type\)](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)

Page: Validate a custom field "Story Points" has been given a value in Fibonacci sequence

Page: Validate compatible values selection among dependent custom fields

Page: Validate only issue links created in transition screen

Page: Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B

Page: Validation and condition based on time expressions

Page: Validation based on the value of a date type project property

Page: Validation on issue attachments

Page: Validation on MIME types of issue attachments

Page: Validation on sibling sub-tasks depending on issue type and status

Page: Validation on the value of a Cascading Select field

- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
 - example
 - post-function
 - custom-field
 - sub-task
- Total of all story points in an epic
 - example
 - custom-field
 - calculated-field
- Show timeliness of an issue based on two date pickers
 - example
 - custom-field
 - calculated-field
- Add and remove a single or a set of items from multi valued fields
 - example
 - post-function
 - custom-field
 - issue-links
 - sub-task
- Highest value of a custom field among linked issues
 - example
 - custom-field
 - calculated-field
- Google Maps location from address
 - example
 - calculated-field
 - custom-field
- Make certain custom field required in resolve screen only if the resolution was set to "Fixed"
 - example
 - validator
 - custom-field