

Limit issue creation per role and issue type

On this page

- [Features used to implement the example](#)
- [Example: Limit issue creation per role and issue type](#)
- [Alternative implementation](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Users in field are \(not\) in a project role](#)
- [Boolean validator with math, date-time or text-string terms](#)

Example: Limit issue creation per role and issue type

A well known limitation of Jira is that it's **not possible** to set **issue creation permissions per Issue Type**. We are going to show how to overcome this limitation by **inserting a validation** in "Create issue" transition in a workflow.

We will show **two different solutions** based on two different features of [Jira Workflow Toolbox](#):

- Using validator [Users in field are \(not\) in a project role](#): this solution is suitable when we have a **different workflows per issue type**, or when we don't want to set different issue creation permissions in a same workflow.
- Using validator [Boolean validator with math, date-time or text-string terms](#): this solution is suitable when we share a workflow among different issue types, and want to set **different issue creation permissions per Issue Type and Project Role in a same workflow**.

Let's suppose that we want to limit issue creation in a workflow, only to user in project roles "**Developers**" or "**Testers**". To do it we will introduce a validator **Users in field are (not) in a project role** in transition "**Create issue**" with the following configuration:

Users in field:	<div>Current user - [User]</div> <div>When a Group Picker or Multi-Group Picker field is selected, all users in the group(s) are evaluated.</div>	
Condition required:	<div><input checked="" type="radio"/> all users in field are at least in one of the selected project roles</div> <div><input type="radio"/> all users in field are in each of the selected project roles</div> <div><input type="radio"/> all users in field are at least out of one of the selected project roles</div> <div><input type="radio"/> all users in field are in none of the selected project roles</div> <div><hr/></div> <div><input type="radio"/> at least one user in field is at least in one of the selected project roles</div> <div><input type="radio"/> at least one user in field is in each of the selected project roles</div> <div><input type="radio"/> at least one user in field is at least out of one of the selected project roles</div> <div><input type="radio"/> at least one user in field is in none of the selected project roles</div>	
Selected project roles:	<div><input type="checkbox"/> Administrators</div> <div><input checked="" type="checkbox"/> Developers</div> <div><input checked="" type="checkbox"/> Testers</div> <div><input type="checkbox"/> Users</div>	
Field must be initialized:	<div><input checked="" type="checkbox"/></div> <div>When checked, if selected field is not initialized the validation will fail.</div>	
Skip validation when: Inhibit the validator under selected circumstances.	<div><input type="checkbox"/> Transition is triggered by a bulk operation.</div> <div><input type="checkbox"/> Transition is triggered by a JIRA Workflow Toolbox post-function.</div> <div><input type="checkbox"/> Current issue is being created by cloning. Only makes sense in <i>Create Issue</i> transition.</div>	
Message to show when validation fails:	<div>Only users with roles "Developers" or "Testers" can create %{00014} issues.</div> <div>Field code injector:<div>Issue type - [Text] - %{00014}</div></div> <div>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.</div> <div>Hide translations for installed languages</div>	
Language	Validation Failure Message Translations	
alemán (Alemania)		
español (España)	Sólo usuarios con los roles "Developers" o "Testers" puede crear incidencias de tipo %{00014}.	
francés (Francia)		
inglés (Estados Unidos)		
inglés (UK)		
japonés (Japón)		

Once configured, transition "**Create issue**" will look like this:

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.

All users in field "**Current user**" are **at least in one** of the following project roles:
"Developers" or **"Testers"**.
 Field **must be** initialized.
 Message to show when validation fails: **"Only users with roles "Developers" or "Testers" can create %{Issue type} issues."**, providing translations for the following languages: **español (España)**. by JWT

Alternative implementation

Let's we implement in a workflow the following issue creation permissions per issue type and project role:

- Issue type **"Bug"** can only be create by users in project roles **"Users"** or **"Testers"**
- Issue type **"Improvement"** can only be create by users in project roles **"Testers"** or **"Product Managers"**
- Issue type **"New Feature"** can only be create by users in project role **"Product Managers"**

We will introduce in transition **"Create issue"** a validator **Boolean validator with math, date-time or text-string terms** with the following configuration:

Boolean expression to be evaluated:

[Line 3 / Col 75] Syntax Specification and Examples ?

```

1 ({00014} = "Bug" IMPLIES (isInRole({00020}, "Testers") OR isInRole({00020}, "Users"))) AND
2 ({00014} = "Improvement" IMPLIES (isInRole({00020}, "Testers") OR isInRole({00020}, "Product Managers"))) AND
3 ({00014} = "New Feature" IMPLIES isInRole({00020}, "Product Managers"))

```

[Logical connectives](#): **and**, **or** and **not**. Alternatively you can also use **&**, **|** and **!**.
[Comparison operators](#): **=**, **!=**, **>**, **>=**, **<** and **<=**. Operators **~**, **!~**, **in**, **not in**, **any in** and **none in** can be used with **strings**, **multi-valued fields** and **lists**.
[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.

Check Syntax

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).

Summary - [Text] - %{00000} ▾

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.

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:

You don't have the project role needed to create %{00014} issues.

Field code injector:

Issue type - [Text] - %{00014} ▾

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](#)

Boolean expression used is:

```
(%{00014} = "Bug" IMPLIES (isInRole({00020}, "Testers") OR isInRole({00020}, "Users"))) AND (%{00014} = "Improvement" IMPLIES (isInRole({00020}, "Testers") OR isInRole({00020}, "Product Managers"))) AND (%{00014} = "New Feature" IMPLIES isInRole({00020}, "Product Managers"))
```

Note that:

- **%{00014}** is code for **"Issue type"**
- **%{00020}** is code for **"Current user"**

Once configured, transition **"Create issue"** will look like this:

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: `{Issue type} = "Bug" IMPLIES (isInRole({Current user}, "Testers") OR isInRole({Current user}, "Users")) AND ({Issue type} = "Improvement" IMPLIES (isInRole({Current user}, "Testers") OR isInRole({Current user}, "Product Managers"))) AND ({Issue type} = "New Feature" IMPLIES isInRole({Current user}, "Product Managers"))`

Message to show when validation fails: **"You don't have the project role needed to create %{Issue type} issues."**

Other examples of that functions

Users in field are (not) in a project role

Page: Limit issue creation per role and issue type

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

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

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
 - example
 - condition
 - validator
 - transition
- Validation based on the value of a date type project property
 - example
 - validator
- Validation and condition based on time expressions
 - example
 - condition
 - validator
 - transition
- Validation on sibling sub-tasks depending on issue type and status
 - example
 - validator
 - sub-task
 - transition
- Validate only issue links created in transition screen
 - example
 - validator
 - issue-links
- Require issue link when resolving as duplicate
 - example
 - validator
 - issue-links
- Validate compatible values selection among dependent custom fields
 - example
 - validator
 - custom-field
- Restrict sub-task type creation depending on parent issue status
 - example
 - validator

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

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

- sub-task
- Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
 - example
 - validator
 - sub-task
- Validation on MIME types of issue attachments
 - example
 - validator
- Validate a custom field "Story Points" has been given a value in Fibonacci sequence
 - example
 - validator
 - custom-field
- Restrict sub-task type creation depending on parent issue type
 - example
 - validator
 - sub-task
- 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
- Block a transition until all sub-tasks have certain fields populated
 - example
 - condition
 - validator
 - sub-task
 - transition