Restrict sub-task type creation depending on parent issue type

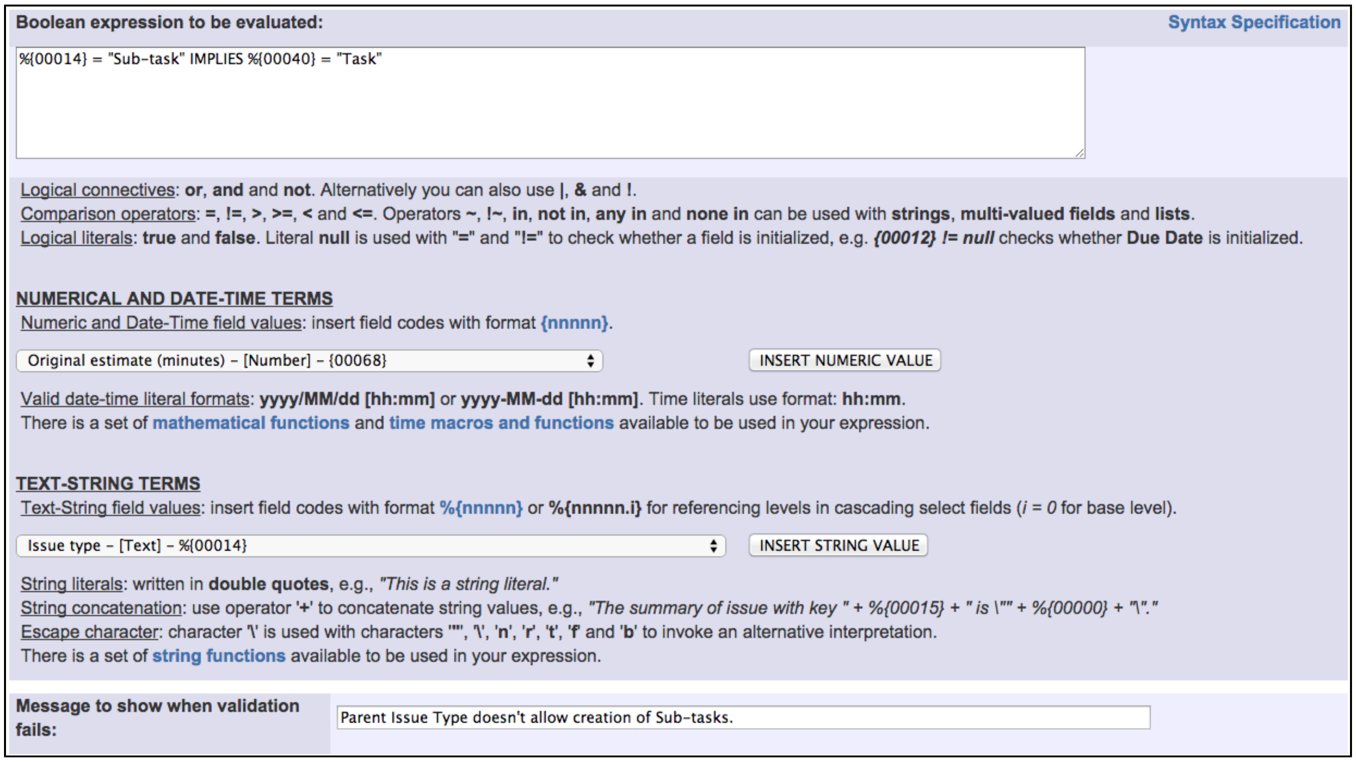
On this page

Features used to implement the example

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

Example: Restrict sub-task type creation depending on parent issue type

I suppose that you have one issue task called "**Sub-task**" and you want to restrict its creation only to "**Task**" parent issue types. In that case you have to add validator to transition "**Create Issue**" of the workflow used by "**Sub-task**", with the following configuration:



Boolean expression to be evaluated: Syntax Specification

```
%{00014} = "Sub-task" IMPLIES %{00040} = "Task"
```

Logical connectives: **or**, **and** 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.

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).

Issue type - [Text] - %{00014} — 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.

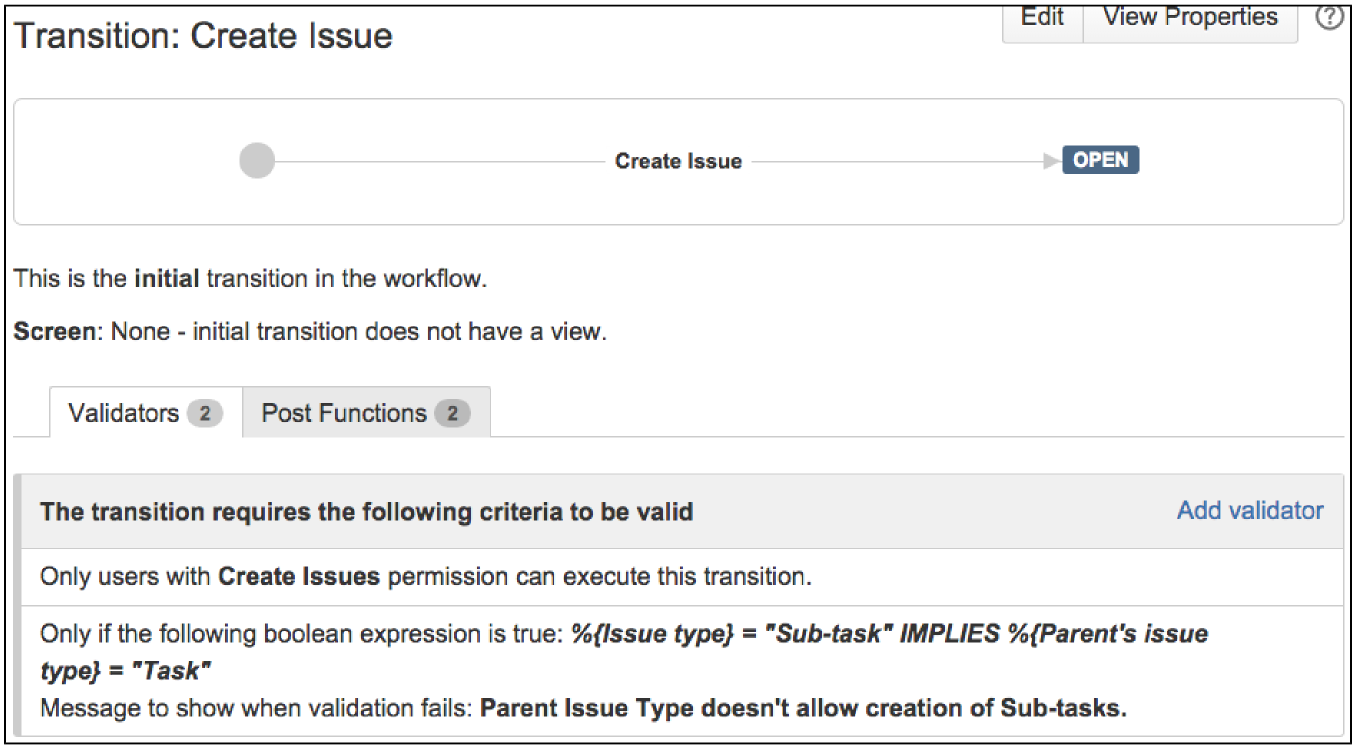
Message to show when validation fails: Parent Issue Type doesn't allow creation of Sub-tasks.

Text to be parsed is: **`%{00014} = "Sub-task" IMPLIES %{00040} = "Task"`**

Note that:

- **%{00014}** is field code for **"Issue type"**
- **%{00040}** is field code for **"Parent's issue type"**
- Logical connective **IMPLIES** is available since version 2.1.22. If you have installed a previous version, you can use the following equivalent expression: **%{00014} != "Sub-task" OR %{00040} = "Task"**

Once configured, the transition **Create Issue** will look like this:



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

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

The transition requires the following criteria to be valid

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

Only if the following boolean expression is true: ***%{Issue type} = "Sub-task" IMPLIES %{Parent's issue type} = "Task"***
Message to show when validation fails: **Parent Issue Type doesn't allow creation of Sub-tasks.**

Other variation of the usage example

Avoid any type of sub-task creation only to parent issue types "Bug" and "Enhancement"

```
%{00041} = null IMPLIES %{00040} not in ["Bug", "Enhancement"]
```

being **%{00041}** field code for field **"Parent's issue key"**

an equivalent expression without using connective **IMPLIES** and operator **NOT IN**:

```
%{00041} != null OR %{00040} != "Bug" AND %{00040} != "Enhancement"
```

Avoid "Sub-task" type creation only to parent issue types "Bug" and "Enhancement"

```
%{00014} = "Sub-task" IMPLIES %{00040} not in ["Bug", "Enhancement"]
```

or an equivalent expression without using connective **IMPLIES** and operator **NOT IN**:

```
%{00041} != "Sub-task" OR %{00040} != "Bug" AND %{00040} != "Enhancement"
```

Limit "Sub-task" type creation only to parent issue type "Task", and "Agile Sub-task" type creation only to parent issue types "Story" and "Epic"

```
(%{00014} = "Sub-task" IMPLIES %{00040} in ["Task"]) AND (%{00014} = "Agile Sub-task" IMPLIES %{00040} in
["Story", "Epic"])
```

or an equivalent expression without using connective **IMPLIES** and operator **IN**:

```
(%{00014} != "Sub-task" OR %{00040} = "Task") AND (%{00014} != "Agile Sub-task" OR %{00040} = "Epic" OR %
{00040} = "Story")
```

Other examples of that function

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
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
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

Related Usage Examples

- Validation on sibling sub-tasks depending on issue type and status
  - example
  - validator
  - sub-task
  - transition
- Restrict sub-task type creation depending on parent issue status
  - example
  - validator
  - sub-task
- Restrict sub-task type creation depending on parent issue type
  - example
  - validator
  - sub-task
- Block a transition until all sub-tasks have certains fields populated
  - example
  - condition
  - validator
  - sub-task
  - transition
- 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
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
  - example
  - post-function
  - custom-field
  - sub-task
- Transition sub-tasks when parent is transitioned
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Transition only a sub-task among several ones
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Add and remove a single or a set of items from multi valued fields
  - example
  - post-function
  - custom-field
  - issue-links
  - sub-task
- Automatically close resolved sub-tasks when parent issue is closed

- example
- post-function
- sub-task
- transition
- outdated

- Change parent's status depending on sub-task's summary
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Close parent issue when all sub-tasks are closed
    - example
    - condition
    - validator
    - post-function
    - sub-task
    - transition