

Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status

On this page

- [Features used to implement the example](#)
- [Example: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Boolean validator with math, date-time or text-string terms](#)
- [Copy a parsed text to a field](#)

Example: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status

If sub-tasks are raised while the parent is at a status of **"Released"**, the parent cannot be moved to status **"Closed"** until all sub-tasks are closed.

Create a **"Text field (single line)"** custom field in sub-task issue called **"Parent status at sub-task's creation"**

Insert post-function [Copy a parsed text to a field](#) into sub-task's **"Create Issue"** with the following configuration:

Target field:

Parent status at subtask's creation - [Text Field (single line)]

Field to be written with the result of the parsed text.

Parsing Mode:

Basic **Basic mode:** Insert field codes anywhere in the text, and they will be replaced with corresponding field values. Field code formats are `%(nnnnn)`, and `%(nnnnn.i)` for Cascading Select fields (i = 0 for base level).

Advanced **Advanced mode:** Strings literals are written in double quotes (*"This is a string."*). Operator `*` is used to concatenate strings, and field codes are like in basic mode, e.g., `"Issue key is " + %(00015) + "."`. More information at [parser syntax documentation](#).

Text to be parsed and then copied to target field: [Syntax Specification](#)

```
1  %{00042}
```

Parent's issue status - [Issue status] - %{00042} Insert String Value

Note that:

- `%(00042)` is field code for **"Parent's issue status"** virtual field
- Beware to introduce that post-function after **"Creates the issue originally"** system post-function

Insert [Boolean validator with math, date-time or text-string terms](#) validator into **"Close issue"** transition in parent's workflow with the following configuration:

Boolean expression to be evaluated: Syntax Specification

```
1 count(filterByPredicate(subtasks(), ^{%11900} = "Released" AND ^{%00016} != "Closed")) = 0
```

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}

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).

Parent status at subtask's creation - [Text Field (single line)] - %{11900}

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:

Boolean expression used is: `count(filterByPredicate(subtasks(), ^{%11900} = "Released" AND ^{%00016} != "Closed")) = 0`

Note that:

- **{11900}** is field code for **"Parent status at subtask's creation"** custom field in my particular JIRA instance. You should replace **11900** with the field code in your Jira instance
- **{00016}** is field code for **"Issue status"**

Other examples of that functions

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"

Related Usage Examples

- Make a sub-task's status match parent issue's current status on creation
 - example
 - post-function
 - sub-task
 - transition
- Create a comment in sub-tasks when parent transitions
 - example
 - post-function
 - sub-task
 - transition
- Transition parent issue only when certain issue sub-task types are done
 - example
 - validator
 - sub-task
 - transition
- Validation on sibling sub-tasks depending on issue type and status
 - example
 - validator

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

Copy a parsed text to a field

Page: Add all assignees of certain sub-task types to a "Multi-User Picker" custom field

Page: Add and remove a single or a set of items from multi valued fields

Page: Add current user to comment

Page: Add or remove request participants

Page: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."

Page: Assign issue based on the value of a Cascading Select custom field

Page: Assign issue to last user who executed a certain transition in the workflow

Page: Automatically close resolved sub-tasks when parent issue is closed

Page: Automatically reopen parent issue when one of its sub-tasks is reopened

Page: Calculate the time elapsed between 2 transition executions

Page: Close parent issue when all sub-tasks are closed

Page: Combine the values of several Multi-User picker fields

Page: Compose a parsed text including the "full name" or a user selected in a User Picker custom field

Page: Compose dynamic text by inserting field values in a text template

Page: Copy issue labels to a custom field

Page: Copy the value of a user property into a user picker

Page: Create a comment in sub-tasks when parent transitions

Page: Execute transition in epic

Page: Getting the number of selected values in a custom field of type Multi Select

Page: Limit the number of hours a user can log per day

Page: Make a sub-task's status match parent issue's current status on creation

Page: Make parent issue progress through its workflow

Page: Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"

Page: Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status

Page: Parse Email addresses to watchers list

Page: Parsing text from last comment and appending it to issue's summary

Page: Remove versions selected in a version picker custom field

Page: Replace certain issue link types with different ones

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

- sub-task
- transition
- Enforce certain type of sub-tasks to be "Resolved" to allow executing a transition
 - example
 - validator
 - sub-task
 - transition
- Make parent issue progress through its workflow
 - example
 - post-function
 - sub-task
 - transition
- Automatically close resolved sub-tasks when parent issue is closed
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Transition only a sub-task among several ones
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Block a transition until all sub-tasks have certain fields populated
 - example
 - condition
 - validator
 - sub-task
 - transition
- Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
 - example
 - validator
 - post-function
 - sub-task
 - transition
- Transition sub-tasks when parent is transitioned
 - 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
- Proceed with a task only when all sub-tasks are completed
 - example
 - condition
 - validator
 - sub-task
 - transition
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Change parent's status depending on sub-task's summary
 - example
 - post-function
 - sub-task
 - transition
 - outdated

Page: Set a Select or Multi-Select field using regular expression to express the values to be assigned

Page: Set assignee depending on issue type

Page: Set field depending on time passed since issue creation

Page: Set priority for issues that have been in a certain status for longer than 24 hours

Page: Set security level based on groups and project roles the reporter or creator are in

Page: Transition linked issues in currently active sprint

Page: Transition only a sub-task among several ones

Page: Transition parent issue only when certain issue sub-task types are done

Page: Update Cascading Select custom field with a value of the field in parent issue

Page: Update checkboxes custom field if a file has been attached during a transition

Page: Validation on issue attachments

Page: Validation on MIME types of issue attachments

Page: Writing a comment to blocked issues when blocking issues are resolved