

Automatically close resolved sub-tasks when parent issue is closed

On this page

- [Features used to implement the example](#)
- [Example: Automatically close resolved sub-tasks when parent issue is closed](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Write field on linked issues or sub-tasks](#)
- [Copy parsed text to a field](#)



This example is considered **outdated**. Please head over to the one using our [Transition issues](#) post function [here](#).

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

Let's see how to automatically close all sub-tasks when parent issue is closed. We do it in two steps:

We copy the name of the target status ("Closed") in auxiliary field **"Ephemeral string 1"** using post-function [Copy parsed text to a field](#):

Target field:

Ephemeral string 1

Text to be parsed and then copied to target field:

Closed

- **Compose a free text** by inserting field codes that will be replaced by corresponding field values prior to be copied to target field.

- You can reference parent and child values of **cascading select fields** writing `%(xxxxx.0)` for parent value, and `%(xxxxx.1)` for child value.

- You can change **reporter**, **assignee**, **due date**, **issue status**, **priority**, **resolution**, **labels**, **components**, **fixed versions**, **affected versions**, **original estimate**, **estimated**, **time spent**, and **security level** by choosing the suitable target field and value to be assigned.

- To assign **cascading selects**, **multi selects**, **multi checkboxes**, **components**, **labels**, **fixed versions** and **affected versions** you should use comma or semicolon separated values.

- Additionally, fields of type **Select list**, **Radio button**, **Multi select**, **Multicheck box**, **Multi user**, **Multi groups**, **Components** and **Versions** can be set through regular expressions: options that matches a regular expression can be set by writing `//(regular_expression)/`, and options that doesn't match a regular expression can be set by writing `/(regular_expression)/`.

- **Setting and unsetting individual values** in multi-valued fields, leaving the rest untouched, can be achieved simply by inserting a character `*` or `^` preceding the value or the list of values. You can insert more than one `*` / `^` character in a sole setting operation.

- Fields **Attachments** (only new attachments will be added) and **Attachments** (all current attachments will be replaced) expect one or more issue keys whose attachments will be copied to current issue.

- You can also use this post-function to **cast a string into a number**.

We use post-function [Write field on linked issues or sub-tasks](#) to copy the value in **"Ephemeral string 1"** into virtual field **"Issue status"** of all the resolved sub-tasks.

Source field to be read in current issue:	Ephemeral string 1
Target field to be written in linked issue or subtask:	Issue status
Filtering by inward issue link type:	☐ is blocked by ☐ is cloned by ☐ is duplicated by ☐ relates to Only issues linked to current issue by selected inward issue link types will be written.
Filtering by outward issue link type:	☐ blocks ☐ clones ☐ duplicates ☐ relates to Only issues linked to current issue by selected outward issue link types will be written.
Write also subtasks fulfilling condition on issue type, status and project:	☒

Filtering linked issues or subtasks by issue type:	☐ Bug ☐ Epic ☐ Improvement ☐ New Feature ☐ Story ☐ Task ☐ Sub-task ☐ Technical task Selected issue types will be written, but if you don't select any, it won't be applied any filter by issue type. In that case all the issue types will be written.
Filtering linked issues or subtasks by status:	☐ Open ☐ In Progress ☐ Reopened ☒ Resolved ☐ Closed Selected statuses will be written, but if you don't select any, it won't be applied any filter by status. In that case issues in any status will be written.
Linked issues or subtasks belong to:	☒ any project ☐ current project ☐ any but current project

Once configured, the transition looks like this:

All
Conditions (2)
Validators (0)
Post Functions (7)

☐ Add a new post function to the unconditional result of the transition.

The following parsed text will be copied to **Ephemeral string 1**:
Closed
Edit | Move Down | Delete

THEN

Value of field **Ephemeral string 1** in current issue will be copied to field **Issue status** in linked issues or subtasks filtering by:
Inward issue link types: **none**
Outward issue link types: **none**
Subtasks fulfilling conditions on issue type, status and project **will be written**.
Issue types: **any**
Statuses: **Resolved**.
Linked issues or subtasks may belong to **any** project.
Edit | Move Up | Move Down | Delete

THEN

Other examples of that functions

Write field on linked issues or sub-tasks

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

Page: Automatically become watcher of every issue blocking an issue assigned to you

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

Page: Automatically resolve an epic when all its stories are resolved

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

Page: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"

Page: Copy attachments from one issue to another

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

Page: Creating a Jira Service Desk internal comment

Page: Creating a Jira Service Desk internal comment on linked issues

Page: Execute transition in epic

Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows

Page: Moving sub-tasks to "Open" status when parent issue moves to "In Progress"

Page: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue

Page: Transition sub-tasks when parent is transitioned

Copy 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

Related Usage Examples

- Validation on sibling sub-tasks depending on issue type and status
 - example
 - validator
 - sub-task
 - transition
- Block a transition until all sub-tasks have certain fields populated
 - example
 - condition
 - validator
 - sub-task
 - transition
- 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
- Automatically close resolved sub-tasks when parent issue is closed
 - example
 - post-function
 - sub-task
 - transition
 - outdated

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

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

- 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
- Proceed with a task only when all sub-tasks are completed
 - example
 - condition
 - validator
 - sub-task
 - transition
- Prevent transitioning when there is a blocking issue
 - example
 - validator
 - issue-links
 - sub-task
 - transition
- Transition parent issue only when certain issue sub-task types are done
 - example
 - validator
 - sub-task
 - transition
- Enforce certain type of sub-tasks to be "Resolved" to allow executing a transition
 - example
 - 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