

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

On this page

- [Features used to implement the example](#)
- [Example: Moving sub-tasks to "Open" status when parent issue moves to "In Progress"](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Update issue fields](#)
- [Write field on linked issues or sub-tasks](#)



This example is considered **outdated**.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

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

We see provision in [Jira Workflow Toolbox](#) to change parent issue's status when a sub-task's status changes. Similarly, is there a way to change a sub-task's status when its' parent issue's status changes? For instance, if parent issue status changes to **"In Progress"**, sub-task status change to **"Open"**.

Yes, you can do it inserting [Write field on linked issues or sub-tasks](#) or [Update issue fields](#) post-function in transition **"Start Progress"** of parent's workflow. In the case of [Update issue fields](#) you should use the following configuration:

Target fields and Source values:
Select the target fields that will be set and the source values for each of them.

Target field:

Summary - [Text]
Add

Add a field to be set in selected issues.

Target Field	Type of Value	Source Value	Don't Overwrite	Actions
Issue status	Parsed text (basic mode)	Open		<a>Edit <a>Remove

Target Issues:
Select the issue(s) to be updated.

Issue Selection Mode:

☐ Current Issue
☐ Parent Issue
☐ Linked Epic
☐ Linked Issues
☒ Subtasks
☐ Sibling Subtasks
☐ Issues under Epic
☐ Sibling issues under Epic
☐ JQL Query
☐ Issue List

Subtasks of current issue. If current issue has no subtasks, nothing will happen.

Issue Type Filter:

Issue types to filter the selected issues. If nothing is selected all issue types are updated.

Issue Status Filter:

Issue statuses to filter the selected issues. If nothing is selected issues in any status are updated.

Resolution Filter:

Resolutions to filter the selected issues. If nothing is selected issues with any resolution are updated.

Additional options:

☐ Enable email notifications on issues to be written, according to applicable notification scheme.
☐ Update issue immediately after field writing. A specific entry will be created in issue history for each field writing.

Conditional execution:
Optional boolean expression that should be satisfied in order to actually execute the post-function.
(Syntax Specification)

1

Leave the field empty for executing the post-function unconditionally.
Collection of Examples
[Line 1 / Col 1]

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

Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ 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.

String Field Code Injector:

Summary - [Text] - %{00000}

Numeric/Date Field Code Injector:

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

Check Syntax

Once configured, the transition will look like this:

The following will be processed after the transition occurs

[Add post function](#)

1. Update the following target issues:

Subtasks of current issue.

Filter the target issues by:

Subtask Issue types: **any**

Statuses: **any**

Resolutions: **any**

Target fields and Source values:

Target Field	Type of Value	Source Value	Don't Overwrite
Issue status	Parsed text (basic mode)	Open	

This feature will be run as user in field **Current user**.

by JWT

Other examples of that functions

Update issue fields

Page: [Add and remove a single or a set of items from multi valued fields](#)

Page: [Compose dynamic text by inserting field values in a text template](#)

Page: [Creating a Jira Service Desk internal comment](#)

Page: [Creating a Jira Service Desk internal comment on linked issues](#)

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: [Parse Email addresses to watchers list](#)

Page: [Set priority for issues that have been in a certain status for longer than 24 hours](#)

Page: [Transition linked issues in currently active sprint](#)

Page: [Transition only a sub-task among several ones](#)

Page: [Using project properties to calculate custom sequence numbers](#)

Page: [Writing a comment to blocked issues when blocking issues are resolved](#)

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](#)

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](#)
- [Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"](#)
 - [example](#)
 - [post-function](#)
 - [sub-task](#)
 - [transition](#)

- outdated
- 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
- 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