

# Transition only a sub-task among several ones

## On this page

- [Features used to implement the example](#)
- [Example: Transition only a sub-task among several ones](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

## Features used to implement the example

- [Update issue fields](#)
- [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This example is considered **outdated**.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: Transition only a sub-task among several ones

Could you please advise how I can transit only first sub-task on parent issue transition.

It will require [JIRA Workflow Toolbox 2.1.28](#) or higher. It can be done using 3 post-functions:

[Copy a parsed text to a field](#) post-function to write the **name of a transition** into virtual field "Ephemeral string 1" virtual field. In this example I use transition "Start Progress":

Target field:

Ephemeral string 1 - [Text]

Field to be written with the resulting parsed text.

☐ Don't overwrite target field if it's already set.

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:

[ Line 1 / Col 16 ][Syntax Specification](#)[Check Syntax](#)

1

Start Progress

Summary - [Text] - %{00000}

Insert String Value

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

Insert Numeric Value

[Copy a parsed text to a field](#) post-function to write the **issue key** of first sub-task order by a certain field into virtual field "**Ephemeral string 2**". In this example I order sub-tasks by **creation date-time** in ascending order:

**Target field:**

Ephemeral string 2 - [Text]

Field to be written with the resulting parsed text.

☐ Don't overwrite target field if it's already set.

**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:**

[ Line 1 / Col 48 ] [Syntax Specification](#) Check Syntax

1 toString(first(sort(subtasks(), {00009}, ASC)))

Summary - [Text] - %{00000}

Insert String Value

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

Insert Numeric Value

Text to be parsed is: `toString(first(sort(subtasks(), {00009}, ASC)))`

Note that:

- **{00009}** is code for numeric value of field "**Date and time of creation**"
- We are using **advanced** parsing mode

[Update issue fields](#) post-function for copying the name of the transition in "**Ephemeral field 1**" into virtual field "**Execute transition**" of issue whose key is in "**Ephemeral field 2**":

**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                                        |
|--------------------|------------------------|--------------------|-----------------|------------------------------------------------|
| Execute transition | Field in current issue | Ephemeral string 1 |                 | <a href="#">Edit</a><br><a href="#">Remove</a> |

**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

1 issuekey in ("%00062")

String Field Code Injector:  
Summary - [Text] - {%00000}

Numeric/Date Field Code Injector:  
Original estimate (minutes) - [Number] - {00068}

[Check Syntax](#)

- Field codes with format %(nnnnn) may be inserted in the JQL Query, and will be replaced with field values at runtime. Most times it's a good idea to write field codes between double quotes (e.g. "%{00001}"), since field values may contain blank spaces that will produce JQL parsing errors at runtime.  
- Cascading Select fields and Multi-level Cascading Select fields specific levels can be referenced with %(nnnnn.0} for parent level, %(nnnnn.1} for child level, etc.

**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](#)

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.

[Check Syntax](#)

String Field Code Injector:  
Summary - [Text] - {%00000}

Numeric/Date Field Code Injector:  
Original estimate (minutes) - [Number] - {00068}

Text to be parsed is: `issuekey in ("%00062")`

Note that:

- `%{00062}` is field code for "Ephemeral string 2"

Once transition is configured, it will look like this:

The following will be processed after the transition occurs

Add post function

1. The following text parsed in **basic** mode will be copied to **Ephemeral string 1**:

**Start Progress**

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

by JWT

2. The following text parsed in **advanced** mode will be copied to **Ephemeral string 2**:

`toString(first(sort(subtasks(), {Date and time of creation}, ASC)))`

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

by JWT

3. Update the following target issues:

**JQL query**

`issuekey in ("%{Ephemeral string 2}")`

Target fields and Source values:

| Target Field       | Type of Value          | Source Value       | Don't Overwrite |
|--------------------|------------------------|--------------------|-----------------|
| Execute transition | Field in current issue | Ephemeral string 1 |                 |

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](#)

### 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](#)  
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](#)

## 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](#)
- [Change parent's status depending on sub-task's summary](#)
  - [example](#)
  - [post-function](#)
  - [sub-task](#)
  - [transition](#)
  - [outdated](#)

[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](#)

- [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](#)
- [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](#)