

Proceed with a task only when all sub-tasks are completed

On this page

- [Features used to implement the example](#)
- [Example: Proceed with a task only when all sub-tasks are completed](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Validation on sub-tasks](#)

Example: Proceed with a task only when all sub-tasks are completed

I have a task and a sub-task. I need to create a condition for that: I only proceed with a task only when all sub-tasks are completed.

Let's suppose you want to prevent a transition "T" from being executed while there exists a sub-task in a status different from "Resolved" or "Closed".

You should insert in transition "T" a [Condition on sub-tasks](#) or [Validation on sub-tasks](#) with the following configuration:

Subtask's issue types:	☐  Sub-task ☐  Technical task Selected issue types will be allowed. Nevertheless, if you didn't select any issue type, and checked "Allow unselected issue types", there won't be applied any filter by issue type, i.e., every issue type will be allowed.
Subtask's statuses:	☐  Open ☐  In Progress ☐  Reopened ☒  Resolved ☒  Closed ☐  Assigned Subtasks in selected statuses will be allowed. Nevertheless, if you didn't select any issue status, and checked "Allow unselected statuses", there won't be applied any filter by status, i.e., every issue status will be allowed.
Subtask's resolutions:	☐ UNRESOLVED, i.e. no resolution value ☐ Fixed ☐ Won't Fix ☐ Duplicate ☐ Incomplete ☐ Cannot Reproduce ☐ Done Subtasks with selected resolutions will be allowed. Nevertheless, if you didn't select any issue resolution, and checked "Allow unselected resolutions", there won't be applied any filter by resolution, i.e., every resolution will be allowed.

Filtering by field values: Optional boolean expression that should be satisfied by issues in order to be selected. (Syntax Specification)	1
Leave field empty for no filtering. <u>Logical connectives:</u> or, and and not. Alternatively you can also use , & and !. <u>Comparison operators:</u> =, !=, >, >=, < and <=. Operators ~, !~, in, not in, any in and none in can be used with strings, multi-valued fields and lists. <u>Logical literals:</u> 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. <u>Numeric/Date Field Code Injector:</u> Original estimate (minutes) - [Number] - {00068} ↕ Field Code for Current Issue Field Code for Foreign Issues <u>String Field Code Injector:</u> Summary - [Text] - %{00000} ↕ Field Code for Current Issue Field Code for Foreign Issues Term <i>"Foreign Issues"</i> refers to issues susceptible to being filtered, i.e., linked issues, subtasks, or any issue different from current one. Example 1: boolean condition {00012} <= ^{00012} will require that issues have <i>Due Date</i> equal or later than current issue's <i>Due Date</i>. Example 2: boolean condition %{00074} ~ ^%{00074} AND ^%{00017} in ["Blocker", "Critical"] will require that issues have <i>Fixed versions</i> contained in current issue's <i>Fixed versions</i> and <i>Priority</i> is <i>Blocker</i> or <i>Critical</i>.	
Minimum required number of subtasks:	0 Minimum number of subtasks required fulfilling selected filtering conditions (issue type, status, resolution and field values).
Maximum allowed number of subtasks:	1000 Maximum number of subtasks allowed fulfilling selected filtering conditions (issue type, status, resolution and field values).
Allow unselected issue types:	☒ Subtasks in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, field values and number. Nevertheless, if none of the issue types are selected, checking this option will make the condition behave as if you had selected every issue type.
Allow unselected issue types:	☒ Subtasks in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, field values and number. Nevertheless, if none of the issue types are selected, checking this option will make the condition behave as if you had selected every issue type.
Allow unselected statuses:	☐ Subtasks in unselected statuses will be ignored, i.e., they will be allowed regardless of their resolution, field values and number. Nevertheless, if none of the statuses are selected, checking this option will make the condition behave as if you had selected every status.
Allow unselected resolutions:	☒ Subtasks in unselected resolutions will be ignored, i.e., they will be allowed regardless of their field values and number. Nevertheless, if none of the resolutions are selected, checking this option will make the condition behave as if you had selected every resolution.
Allow unsatisfied condition on field values:	☐ Subtasks not satisfying filter by field values will be ignored, i.e., they will be allowed regardless of their number.
Message to show when validation fails:	All subtasks should be "Resolved" or "Closed" before executing current transition. Set translations for installed languages
Add Cancel	

Once configured, transition "T" will look like this:

Triggers 0
Conditions 0
Validators 1
Post Functions 6

The transition requires the following criteria to be valid
Add validator

Validation on subtasks:

At least 0 and no more than 1000 subtasks with the following characteristics:

- Issue types: **any**
- Statuses: **Resolved** or **Closed**.
- Resolutions: **any**

About the rest of subtasks:

- Unselected statuses are **not allowed**.

Message to show when validation fails: **"All subtasks should be "Resolved" or "Closed" before executing current transition."**

Other examples of that function

Page: [Close parent issue when all sub-tasks are closed](#)
Page: [Enforce certain type of sub-tasks to be "Resolved" to allow executing a transition](#)
Page: [Make linked issues, sub-tasks and JQL selected issues progress through its workflows](#)
Page: [Make parent issue progress through its workflow](#)
Page: [Proceed with a task only when all sub-tasks are completed](#)
Page: [Transition parent issue only when certain issue sub-task types are done](#)

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](#)
- [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