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

On this page

- Features used to implement the example
- Example: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
- Other examples of that functions
- Related Usage Examples

Features used to implement the example

- **Read fields from linked issues or sub-tasks**
- **Write field on linked issues or sub-tasks**

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

Issue type A is linked to **issue type B**. Issue A has sub-tasks and people log work at the sub-task level. When **issue A** is closed I need all work logged in its sub-tasks to be copied to **issue B**. Actually it has to be added since issue B may have work logged from other issues (Issue B is a summary of what is being logged at a lower level). I have been trying using virtual field Total time spent (minutes) in Close post function with copy parsed value and write to linked issues, but it doesn't work...maybe there is a conversion that I am missing.

It is possible to implement your requirements using post-functions **Read fields from linked issues or sub-tasks** and **Write field on linked issues or sub-tasks**, and virtual field "**Total time spent (minutes)**".

Let's suppose that issue A is linked to issue B using issue link type "**blocks**", i.e., we would read "**Issue A blocks issue B**". We are going to implement in workflow of issue A a behavior associated to transition "**Close Issue**", that will add to issue B all time spent (work logs) in sub-tasks of issue A, and also the time spent of issue A (work directly logged into issue A).

Use post-function **Read fields from linked issues or sub-tasks** with the following configuration:

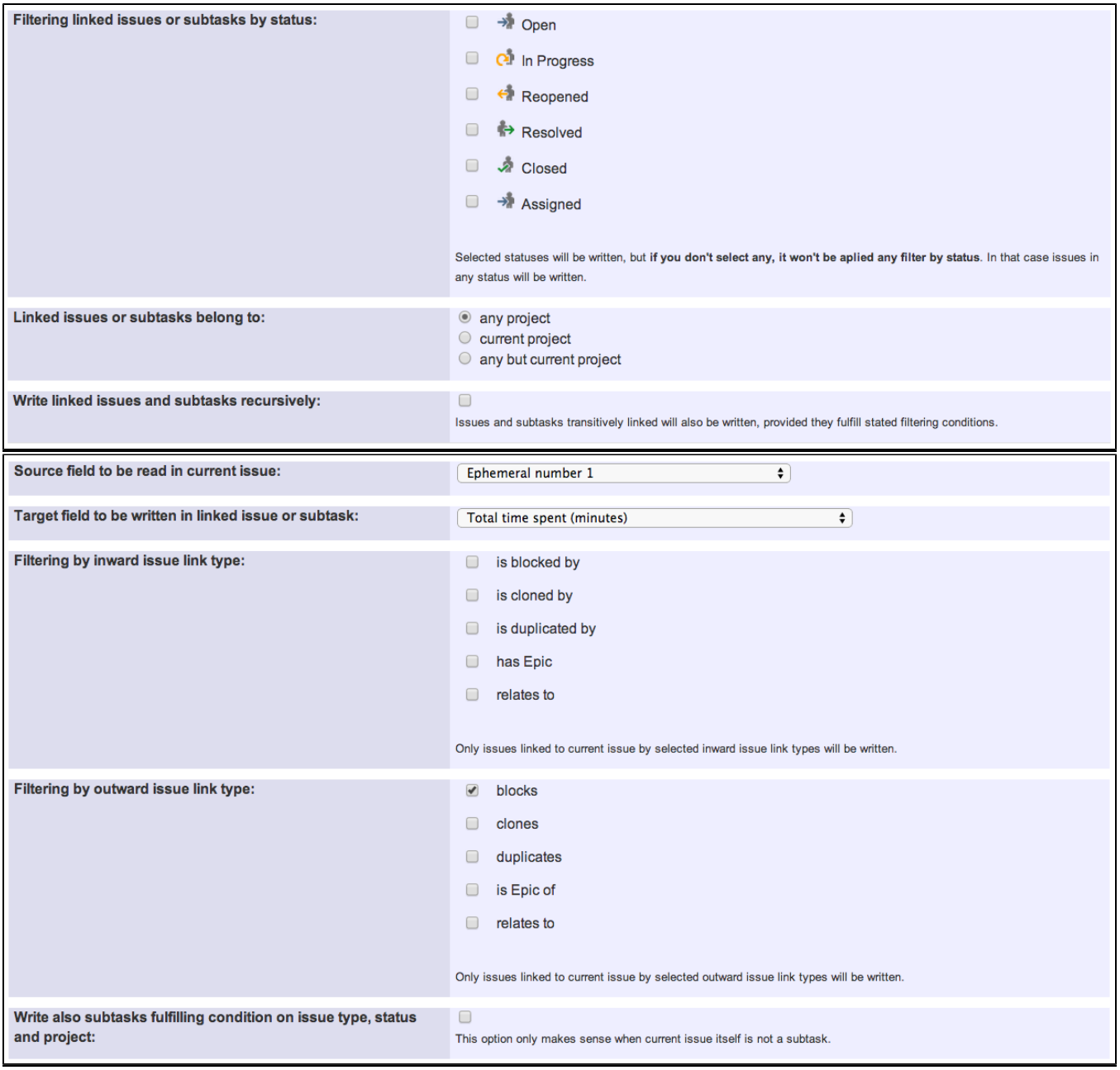
Source field to be read from linked issues or subtasks:	Total time spent (minutes)
Target field to be written in current issue:	Ephemeral number 1
Value to write in target field in case source field is of type Number, Priority, Date, Date and Time:	sum of values
Filtering by inward issue link type:	☐ is blocked by ☐ is cloned by ☐ is duplicated by ☐ has Epic ☐ relates to Only issues linked to current issue by selected inward issue link types will be read.
Filtering by outward issue link type:	☑ blocks ☐ clones ☐ duplicates ☐ is Epic of ☐ relates to Only issues linked to current issue by selected outward issue link types will be read.

Read also subtasks fulfilling condition on issue type, status and project:	☑ This option only makes sense when current issue itself is not a subtask.
Read also sibling subtasks fulfilling condition on issue type, status and project:	☐ Sibling subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.
Filtering linked issues or subtasks by issue type:	☐ Bug ☐ Improvement ☐ New Feature ☐ Task ☐ gh.issue.epic ☐ gh.issue.story ☐ Sub-task ☐ gh.issue.task Selected issue types will be read, but **if you don't select any, it won't be aplied any filter by issue type**. In that case all the issue types will be read.

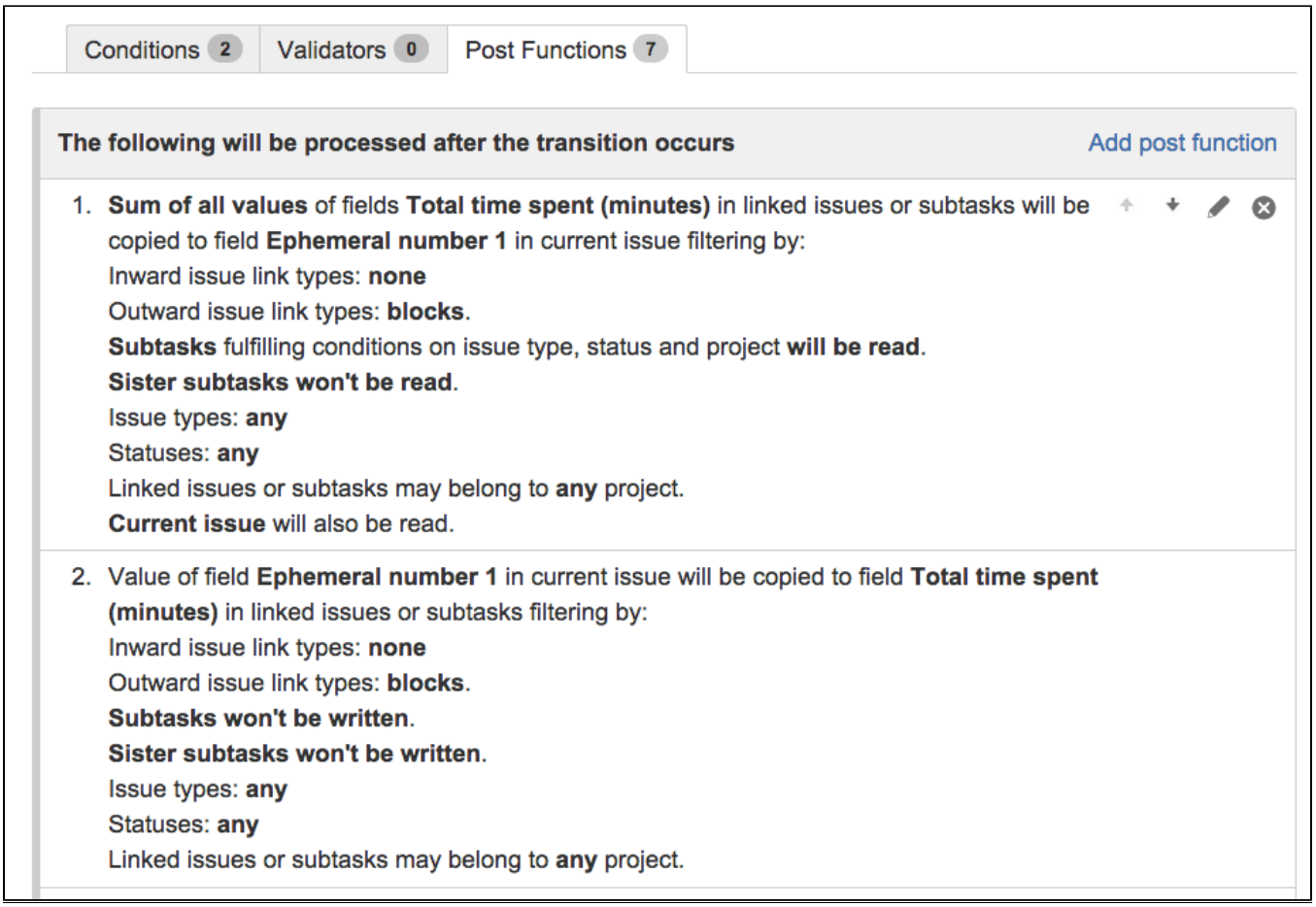
Filtering linked issues or subtasks by status:	☐ Open ☐ In Progress ☐ Reopened ☐ Resolved ☐ Closed ☐ Assigned Selected statuses will be read, but **if you don't select any, it won't be aplied any filter by status**. In that case issues in any status will be read.
Linked issues or subtasks belong to:	◉ any project ○ current project ○ any but current project
Read linked issues and subtasks recursively:	☐ Issues and subtasks transitively linked will also be read, provided they fulfill stated filtering conditions.
Read also current issue:	☑ Current issue will be also read. This option is useful when you want to make the reading cumulative, i.e., you don't want to get current value to be simply overwritten, but enriched with the values of linked issues, or when you want to get current issue's value to be taken into account in a formula (maximum, minimum, average, etc.).

Use post-function **Write field on linked issues or sub-tasks** with the following configuration:

Write also sibling subtasks fulfilling condition on issue type, status and project:	☐ Sibling subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.
Filtering linked issues or subtasks by issue type:	☐ Bug ☐ Improvement ☐ New Feature ☐ Task ☐ gh.issue.epic ☐ gh.issue.story ☐ Sub-task ☐ gh.issue.task Selected issue types will be written, but **if you don't select any, it won't be aplied any filter by issue type**. In that case all the issue types will be written.



Once configured, transition will look like this:



Other examples of that functions

Read fields from linked issues or sub-tasks

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: 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: Make an issue inherit highest priority among those of linked issues
Page: Propagate highest priority from blocked issues to blocking issues
Page: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue

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

Related Usage Examples

- Add and remove a single or a set of items from multi valued fields
  - example
  - post-function
  - custom-field
  - issue-links
  - sub-task
- Prevent transitioning when there is a blocking issue
  - example
  - validator
  - issue-links
  - sub-task
  - transition
- Make linked issues, sub-tasks and JQL selected issues progress through its workflows
  - example
  - condition
  - validator
  - post-function
  - issue-links
  - sub-task
  - transition
- Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC"
  - example
  - post-function
  - issue-links
  - sub-task
  - work-log
- Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
  - example

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

    - post-function
    - issue-links
    - sub-task
    - work-log
- Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
    - example
    - post-function
    - custom-field
    - issue-links
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
    - example
    - post-function
    - custom-field
    - sub-task
- Add all assignees of certain sub-task types to a "Multi-User Picker" custom field
    - example
    - post-function
    - custom-field
    - sub-task
- Create a sub-task for each user selected in a Multi-User Picker
    - example
    - post-function
    - custom-field
    - sub-task
- Update Cascading Select custom field with a value of the field in parent issue
    - example
    - post-function
    - custom-field
    - sub-task
- Validate only issue links created in transition screen
    - example
    - validator
    - issue-links
- Require issue link when resolving as duplicate
    - example
    - validator
    - issue-links
- Ensure that all issues linked with a certain issue link type have "Due Date" field set
    - example
    - validator
    - issue-links
- Block an epic's transition depending on linked issues status and due date
    - example
    - validator
    - issue-links
    - transition
- Writing a comment to blocked issues when blocking issues are resolved
    - example
    - post-function
    - issue-links