

# Log absence time on another issue

## On this page

- [Features used to implement the example](#)
- [Example: Log absence time on another issue](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

## Features used to implement the example

- [Log work](#)
- 

## Example: Log absence time on another issue

In this example, work should be logged on another issue than the current issue where the post function is triggered.

In this specific case, we want to log 8 hours for a custom day on a specific employee issue.

If a user creates an issue called "Absence" with selecting the date for the Absence (custom date picker field called "**Absence Day**") and the issue get's approved, 8 hours should be logged at that date on the employee issue.

The configuration of the workflow transition for the "Absence" issue will look like this:

<b>Target Issues:</b> Select the issue(s) to be logged.	<b>Issue Selection Mode:</b> <div> <input type="radio"/> Current Issue         <input type="radio"/> Parent Issue         <input type="radio"/> Linked Epic         <input type="radio"/> Linked Issues         <input type="radio"/> Subtasks         <input type="radio"/> Sibling Subtasks         <input type="radio"/> Issues under Epic         <input type="radio"/> Sibling Issues under Epic         <input type="radio"/> JQL Query         <input checked="" type="radio"/> Issue List       </div> <div>[ Line 1 / Col 1 ]</div> <div>1 issuesFromJQL("summary ~ \"Employee \" + %{00006}+\"\\")</div> <div> <b>String Field Code Injector:</b>          Start typing to get the list of available fields       </div> <div> <b>Numeric/Date Field Code Injector:</b>          Start typing to get the list of available fields       </div> <div>         Input an expression that returns an issue lists. (<a href="#">Syntax Specification</a>)         <a href="#">Collection of Examples</a> </div>
<b>Worklog author:</b>	Current user - (User) - [00020]
<b>Remaining estimate adjustment:</b>	<input checked="" type="radio"/> Auto adjustment <input type="radio"/> Retain remaining estimate
<b>Time to be logged (minutes):</b>  <a href="#">Syntax Specification</a>	<div>1 480</div> <div>[ Line 1 / Col 1 ]</div> <div> <b>Numeric/Date-Time field code injector:</b>          Start typing to get the list of available fields       </div> <div>         Formula expressing a value in minutes. You can insert fields of type <b>Number</b>, <b>Date</b> and <b>Date-Time</b> using format {nnnnn}. Fields of type Date and Date-Time represent milliseconds elapsed since January 1, 1970, 00:00:00 GMT. <b>Date subtraction allows you to calculate the time elapsed between two dates in milliseconds.</b> To convert milliseconds to minutes you have to <b>divide by (MINUTE)</b>. Example: formula ((00012)-(00009))/ (MINUTE) represents minutes elapsed from issue Creation to Due Date. You can use function <code>subtractDatesSkippingWeekends(minuend_date, subtrahend_date, time_zone)</code> to skip weekends.       </div>
<b>Starting Date-Time:</b>  <a href="#">Syntax Specification</a>	<div>1 stringToDate((%{10201} + " 08:00"), "yyyy-MM-dd hh:mm")</div> <div>[ Line 1 / Col 1 ]</div> <div> <b>Numeric/Date-Time field code injector:</b>          Start typing to get the list of available fields       </div> <div>         Time expression for the starting date-time of the work log entry. In case no expression is entered, <b>current date-time</b> is used. You can insert fields of type <b>Number</b>, <b>Date</b> and <b>Date-Time</b> using format {nnnnn}. Fields of type Date and Date-Time represent milliseconds elapsed since January 1, 1970, 00:00:00 GMT.       </div>
<b>Log comment:</b>	<div>1 Absence</div> <div> <b>Field code injector:</b>          Start typing to get the list of available fields       </div> <div>         - <b>Compose free text</b> by inserting field codes (%{nnnnn}) that will be replaced by corresponding field values prior to be copied to target field.          - You can insert parent and child values of <b>cascading select fields</b> writing %{nnnnn.0} for parent value, and %{nnnnn.1} for child value.          - You can also insert values of <b>multi-level cascading select fields</b> writing %{nnnnn.1}, with 1 being the level to be read (root level is 0).       </div>
<b>Conditional execution:</b> Optional boolean expression that should be satisfied in order to actually execute the post-function. <a href="#">(Syntax Specification)</a>	<div>1 %{00000} ~ "Absence"</div> <div>[ Line 1 / Col 1 ]</div> <div>         Leave the field empty for executing the post-function unconditionally.         <a href="#">Collection of Examples</a> </div> <div> <b>Logical connectives:</b> and, or and not. Alternatively you can also use &amp;,   and !.  <b>Comparison operators:</b> =, !=, &gt;, &gt;=, &lt; and &lt;=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multi-valued fields and lists.  <b>Logical literals:</b> 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.       </div> <div> <b>String Field Code Injector:</b>          Start typing to get the list of available fields       </div> <div> <b>Numeric/Date Field Code Injector:</b>          Start typing to get the list of available fields       </div>

Text to be parsed is:

Target issue "Issue List": `issuesFromJQL("summary ~ \"Employee \" + %{00006}+\"\\")`

Starting Date-Time: `stringToDate((%{10201} + " 08:00"), "yyyy-MM-dd hh:mm")`

Note that:

- `%{00006}` is field code for field "Reporter"
- `%{10201}` is field code for custom field "Absence Day"
- Time to be logged has to be entered in minutes

Once configured, the configuration will look like this:

The following will be processed after the transition occurs	Add post function
1. The <b>Resolution</b> of the issue will be set to <b>Done</b> .	
2. Set issue status to the linked status of the destination workflow step.	
3. Add a comment to an issue if one is entered during a transition.	
4. Update change history for an issue and store the issue in the database.	
5. Re-index an issue to keep indexes in sync with the database.	
6. Fire a <b>Generic Event</b> event that can be processed by the listeners.	
7. User in field <b>Current user</b> will log 480 minutes with <b>auto adjustment of remaining estimate</b> adding the following comment: " <b>Absence</b> ", with the following start date: <code>stringToDate( ({Absence Day} + " 08:00"), "yyyy-MM-dd hh:mm")</code> Post-function will only be executed if the following boolean expression is satisfied: <code>%{Summary} ~ "Absence"</code>	<div>by JWT</div>

Result screenshots of post function "Log absence time on another issue"

## Other examples of that function

Page: [Automatic work log with start and stop work transitions](#)

Page: [Log absence time on another issue](#)

Page: [Log absence time on another issue](#)

## Related Usage Examples

- [Limit the number of hours a user can log per day](#)
  - [example](#)
  - [validator](#)
  - [post-function](#)
  - [work-log](#)
- [Make "Time Spent" field required when there is no time logged in the issue](#)
  - [example](#)
  - [validator](#)
  - [work-log](#)
- [Limit valid dates for work logs](#)
  - [example](#)
  - [validator](#)
  - [work-log](#)
- [Sum sub-task's "Time Spent" \(work logs\) and add it to a certain linked issue](#)
  - [example](#)
  - [post-function](#)
  - [issue-links](#)
  - [sub-task](#)
  - [work-log](#)
- [Log absence time on another issue](#)
  - [example](#)
  - [post-function](#)
  - [work-log](#)
- [Set "Total time spent" to "Current date and time - date and time of last update"](#)
  - [example](#)
  - [post-function](#)
  - [work-log](#)
- [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](#)
- [Automatic work log with start and stop work transitions](#)
  - [example](#)
  - [post-function](#)
  - [work-log](#)
- [Automatically log work time when the user uses a "Stop Progress" transition](#)
  - [example](#)

- post-function
- custom-field
- work-log