

Automatically log work time when the user uses a "Stop Progress" transition

On this page

- [Features used to implement the example](#)
- [Example: Automatically log work time when the user uses a "Stop Progress" transition](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Mathematical and date-time expression calculator](#)

Example: Automatically log work time when the user uses a "Stop Progress" transition

Is it possible to log work using a post-function? Are there any other means of achieving this?

What we want to achieve is to automatically log work time when the user presses the Stop Progress button, either by automatically logging the work time or by setting the time in the Work Time field in the stop progress screen.

Yes, you can do it. You have to use the following elements:

- Virtual field "**Total time spent (minutes)**". Writing into this virtual field allows you to log time
- Virtual field "**Current date and time**"
- An auxiliary custom field of type **Date Time** to store the time where you start progress. In the example I call this field "**Start Progress Time**"
- Post-function [Mathematical and date-time expression calculator](#) to do all the calculations and assignments into virtual field "**Total time spent (minutes)**"

We only need to insert post-function [Mathematical and date-time expression calculator](#) into two transitions: "**Start Progress**" and "**Stop Progress**".

Transition "Start Progress": We insert post-function [Mathematical and date-time expression calculator](#) in transition "**Start Progress**" to store into custom field "**Start Progress Time**" the exact moment when you begin working on the issue. This custom field should not be present in edit screen since you only want it to be modified by this post-function. Let's see in screenshots how to do it:

The screenshot shows a configuration window with two main sections. The top section is labeled "Target field:" and contains a dropdown menu with "Start Progress Time" selected. The bottom section is labeled "Formula:" and contains a text area with the text "{00057}" entered.

Once configured, the transition will look like this:

All Conditions (0) Validators (0) Post Functions (7)

■ Add a new post function to the unconditional result of the transition.

Field **Start Progress Time** will be assigned with the calculation result of the formula:
{Current date and time}
 Edit | Move Down | Delete

— THEN

Transition "Stop Progress": We insert post-function **Mathematical and date-time expression calculator** in transition **"Stop Progress"** to calculate the true time passed since **"Start Progress"** to **"Stop Progress"**, and add this time to the value already present in virtual field **"Total time spent (minutes)"**. Let's see how to do it:

Target field:
 Total time spent (minutes)

Formula:
 {{00025}}+({{00057}}-{{11400}})/{{MINUTE}}

Once configured, the transition will look like this:

All Conditions (1) Validators (0) Post Functions (7)

■ Add a new post function to the unconditional result of the transition.

Field **Total time spent (minutes)** will be assigned with the calculation result of the formula:
{{Total time spent (minutes)}}+({{Current date and time}}-{{Start Progress Time}})/{{MINUTE}}
 Edit | Move Down | Delete

— THEN

This solution trims seconds passed since last whole minute, that is, if time passed since **"Start Progress"** to **"Stop Progress"** were 35 minutes 22 seconds, logged time would be 35 minutes. I could add a new virtual field **"Total time spent (seconds)"** to enable a resolution up to seconds.

Since version 2.2.39 you can do the time calculation considering only a particular work schedule, i.e., only the days and times when actual work is done. To do it you should use the following formula instead:

```
timeDifference({00057}, {nnnnn}, "my_schedule", LOCAL) / {MINUTE}
```

Where **my_schedule** is the name of your particular work schedule. You define your schedule using **Schedules Definition Grammar** at Administration > Add-ons > JIRA WORKFLOW TOOLBOX > Schedule.

Other examples of that function

Page: [Automatic work log with start and stop work transitions](#)
Page: [Automatically log work time when the user uses a "Stop Progress" transition](#)
Page: [Calculate the time elapsed between 2 transition executions](#)
Page: [Getting the number of selected values in a custom field of type Multi Select](#)
Page: [Implement a form with a series of questions and calculate a numeric value based on the answers](#)
Page: [Increment a field or set to 1 if it's not set](#)
Page: [Set "Date-Time Picker" custom field with current date-time](#)
Page: [Set "Due date" 6 natural days \(or work days\) earlier than a "Date Picker" custom field](#)
Page: [Set "Due date" to a specific day of next week no matter of date of creation this week](#)
Page: [Set "Due date" with certain time offset from current date](#)
Page: [Set "Total time spent" to "Current date and time - date and time of last update"](#)
Page: [Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field](#)
Page: [Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC"](#)
Page: [Triage Jira Service Desk email requests \(Move issues\)](#)
Page: [Using project properties to calculate custom sequence numbers](#)

Related Usage Examples

- [Automatically log work time when the user uses a "Stop Progress" transition](#)
 - [example](#)
 - [post-function](#)
 - [custom-field](#)
 - [work-log](#)
- [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](#)
- [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](#)
- [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](#)
- [Validate compatible values selection among dependent custom fields](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Validate a custom field "Story Points" has been given a value in Fibonacci sequence](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Validation on the value of a Cascading Select field](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Make different fields mandatory depending on the value of a Select List custom field](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)
- [Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"](#)
 - [example](#)
 - [validator](#)
 - [custom-field](#)

