

Automatic work log with start and stop work transitions

On this page

- [Features used to implement the example](#)
- [Example: Automatic work log with start and stop work transitions](#)
- [Alternative implementation](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

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

To implement this functionality we will use **two custom fields** and **two post-functions**

- Create a Date Time Picker type custom field called **"Work Start Time"**. In this example the field code for this custom field is **{10002}**.
- virtual field **"Current date and time"**, the field code is **{00057}**.
- Use post-function [Mathematical and date-time expression calculator](#) in transition **"Start Progress"** in order to store current date and time in custom field **"Work Start Time"**.
- Use post-function [Log work](#) in transition **"Stop Progress"** in order to insert a work log for the minutes elapsed between date-time stored in custom field **"Work Start Time"** and current date and time.

Example: Automatic work log with start and stop work transitions

We want to automatically insert a work log for the time passed since triggering of **"Start Progress"** transition to triggering of **"Stop Progress"** transition.

Insert post-function [Mathematical and date-time expression calculator](#) into transition **"Start Progress"** to set custom field **"Work Start Time"** with the value of virtual field **"Current date and time"**:

Target field:

Work Start Time - [Date Time Picker]

Field to be written with the result of evaluation of the formula.

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

Formula:

[Line 1 / Col 9]

Syntax Specification

Check Syntax

1

{00057}

NUMERICAL AND DATE-TIME TERMS

Numeric and Date-Time field values:

insert field codes with format {nnnnn}.

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

Insert Numeric Value

Valid date-time literal formats:

yyyy/MM/dd [hh:mm] or yyyy-MM-dd [hh:mm].

Time literals use format: hh:mm.

There is a set of [mathematical functions](#) and [time macros and functions](#) available to be used in your expression.

TEXT-STRING TERMS

Text-String field values:

insert field codes with format %{nnnnn} or %{nnnnn.i} for referencing levels in cascading select fields (*i* = 0 for base level).

Summary - [Text] - %{00000}

Insert String Value

String literals:

written in **double quotes**, e.g., "This is a string literal."

String concatenation:

use operator '+' to concatenate string values, e.g., "The summary of issue with key " + %{00015} + " is \" + %{00000} + "\".

Escape character:

character '\ ' is used with characters '"', '\', 'n', 'r', 't', 'f' and 'b' to invoke an alternative interpretation.

There is a set of [string functions](#) available to be used in your expression.

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

[Line 1 / Col 1]

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.

String Field Code Injector:

Summary - [Text] - %{00000}

Numeric/Date Field Code Injector:

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

Check Syntax

Run as:

Select the user that will be used to execute this feature. JIRA will apply restrictions according to the permissions, project roles and groups of the selected user.

Current user

User defined by a field.

Input a specific user.

Note that:

- {00057} is field code for numeric value of virtual field "Current date and time"

Once configured, transition "Start Progress" looks like this:

TO DO

Start Progress

IN PROGRESS

Screen: None - it will happen instantly

Triggers 0

Conditions 1

Validators 1

Post Functions 18

The following will be processed after the transition occurs

[Add post function](#)

1. Field **Work Start Time** will be assigned with the calculation result of the formula:

{Current date and time}

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

Insert post-function [Log work](#) into transition "Stop Progress", using a time formula to calculate the number of minutes elapsed between "Start Progress" and "Stop Progress" transitions:

Worklog author:	Current user - [User] ?
Remaining estimate adjustment:	☒ Auto adjustment ☐ Retain remaining estimate
Time to be logged (minutes):	1 <code>{00057} - {10300} / {MINUTE}</code> Syntax Specification Numeric/Date-Time field code injector: Work Start Time - [Date Time Picker] - {10300} Formula expressing a value in minutes. You can insert fields of type Number, Date and Date-Time using format <code>nnnnn</code>. Fields of type Date and Date-Time represent milliseconds elapsed since January 1, 1970, 00:00:00 GMT. Date subtraction allows you to calculate the time elapsed between two dates in milliseconds. To convert milliseconds to minutes you have to divide by {MINUTE}. Example: formula <code>{00012} - {00009} / {MINUTE}</code> 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. Check Syntax
Log comment:	1 <code>Work started at %{10300} and stopped at %{00057}.</code> Field code injector: Work Start Time - [Date Time Picker] - %{10300} - Compose free text by inserting field codes <code>{nnnnn}</code> that will be replaced by corresponding field values prior to be copied to target field. - You can insert parent and child values of cascading select fields writing <code>{nnnnn.0}</code> for parent value, and <code>{nnnnn.1}</code> for child value. - You can also insert values of multi-level cascading select fields writing <code>{nnnnn.1}</code>, with 1 being the level to be read (root level is 0).
Conditional execution:	1 Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification) Leave the field empty for executing the post-function unconditionally. Collection of Examples Logical connectives: and, or and not. Alternatively you can also use <code>&</code>, <code> </code> and <code>!</code>. Comparison operators: <code>=</code>, <code>!</code>, <code>></code>, <code>>=</code>, <code><</code> and <code><=</code>. Operators <code>in</code>, <code>not in</code>, <code>any in</code>, <code>none in</code>, <code>~</code> and <code>!~</code> can be used with <i>strings</i>, <i>multi-valued fields</i> and <i>lists</i>. Logical literals: <code>true</code> and <code>false</code>. Literal <code>null</code> is used with <code>=</code> and <code>!=</code> to check whether a field is initialized, e.g. <code>{00012} != null</code> checks whether <i>Due Date</i> is initialized. Check Syntax
	String Field Code Injector: Summary - [Text] - %{00000} Numeric/Date Field Code Injector: Original estimate (minutes) - [Number] - {00068}

Math expression for obtaining the time elapsed in minutes: $(\{00057\} - \{10300\}) / \{MINUTE\}$

Note that:

- **{00057}** is field code for numeric value of virtual field "Current date and time"
- **{10300}** is field code for numeric value of custom field "Work Start Time". This code is dependent on each particular Jira instance
- When we want a string representation of a date-time field, we use the **string value** field code, i.e., **%{00057}** and **%{10300}**

Once configured, transition "Stop Progress" looks like this:

The screenshot shows the configuration for the 'Stop Progress' transition in a Jira workflow. At the top, a visual representation of the workflow shows a yellow 'IN PROGRESS' status transitioning to a blue 'TO DO' status via the 'Stop Progress' transition. Below this, the 'Screen' is set to 'None - it will happen instantly'. A tabbed interface shows 'Triggers' (0), 'Conditions' (1), 'Validators' (0), and 'Post Functions' (6). The 'Post Functions' tab is active, displaying a list of functions to be processed after the transition. The first function is: '1. User in field **Current user** will log $(\{Current\ date\ and\ time\} - \{Work\ Start\ Time\}) / \{MINUTE\}$ minutes with auto adjustment of remaining estimate adding the following comment: "Work started at %{Work Start Time} and stopped at %{Current date and time}." A link 'Add post function' is visible in the top right of the post functions list.

Alternative implementation

Support for Custom Schedules (since version 2.2.39)

You can also implement the above use case but only considering the actual work time in your organization. To do it you should define your custom work-schedule using [Schedules Definition Grammar](#) at Administration > Add-ons > JIRA WORKFLOW TOOLBOX > Schedule

Let's consider the following work-schedule called "my_work_schedule":

```
# Winter Schedule
MON - THU {
  08:00 - 15:00,
  16:00 - 20:00;
}

FRI {
  08:00 - 15:00;
}

# Summer Schedule
JUN/15 - SEP/15 {
  MON - FRI {
    08:00 - 14:30;
  }
}

# Annual Holidays
JAN/1, MAY/1, NOV/1, DEC/25 {};

# 2017 Holidays
2017/JAN/12, 2017/APR/13, 2017/APR/14, 2017/NOV/23 {};
```

In order to use this custom work-schedule we should use the following configuration in post-function [Log work](#):

Worklog author:	Current user - [User] ?
Remaining estimate adjustment:	☒ Auto adjustment ☐ Retain remaining estimate
Time to be logged (minutes):	1 <code>timeDifference({00057}, {10300}, "my_work_schedule", LOCAL) / {MINUTE}</code> Syntax Specification Numeric/Date-Time field code injector: [Line 1 / Col 72] Original estimate (minutes) - [Number] - {00068} ▼ Formula expressing a value in minutes. You can insert fields of type Number, Date and Date-Time using format <code>nnnnn</code>. Fields of type Date and Date-Time represent milliseconds elapsed since January 1, 1970, 00:00:00 GMT. Date subtraction allows you to calculate the time elapsed between two dates in milliseconds. To convert milliseconds to minutes you have to divide by {MINUTE}. Example: formula <code>{00012} - {00009} / {MINUTE}</code> 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. Check Syntax
Log comment:	1 <code>Work started at %{10300} and stopped at %{00057}.</code> Field code injector: [Line 1 / Col 50] Summary - [Text] - %{00000} ▼ - Compose free text by inserting field codes <code>{%nnnnn}</code> that will be replaced by corresponding field values prior to be copied to target field. - You can insert parent and child values of cascading select fields writing <code>{%nnnnn.0}</code> for parent value, and <code>{%nnnnn.1}</code> for child value. - You can also insert values of multi-level cascading select fields writing <code>{%nnnnn.l}</code>, with <code>l</code> being the level to be read (root level is 0).
Conditional execution:	1 Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification) Leave the field empty for executing the post-function unconditionally. Collection of Examples [Line 1 / Col 1] <u>Logical connectives</u>: and, or and not. Alternatively you can also use <code>&</code>, <code> </code> and <code>!</code>. <u>Comparison operators</u>: <code>=</code>, <code>!=</code>, <code>></code>, <code>>=</code>, <code><</code> and <code><=</code>. Operators <code>in</code>, <code>not in</code>, <code>any in</code>, <code>none in</code>, <code>~</code> and <code>!~</code> can be used with <i>strings</i>, <i>multi-valued fields</i> and <i>lists</i>. <u>Logical literals</u>: <code>true</code> and <code>false</code>. Literal <code>null</code> is used with <code>=</code> and <code>!=</code> to check whether a field is initialized, e.g. <code>{00012} != null</code> checks whether <i>Due Date</i> is initialized. Check Syntax <u>String Field Code Injector:</u> Summary - [Text] - %{00000} ▼ <u>Numeric/Date Field Code Injector:</u> Original estimate (minutes) - [Number] - {00068} ▼

Formula is: `timeDifference({00057}, {10300}, "my_work_schedule", LOCAL) / {MINUTE}`

Other examples of that function

Log work

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

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

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

Mathematical and date-time expression calculator

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](#)

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](#)
- [Set "Total time spent" to "Current date and time - date and time of last update"](#)
 - [example](#)
 - [post-function](#)

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

- 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
- 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
- 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