

Log work

This function has been **renamed** with the **JWT 3.0** release.

Find the new documentation at:

[Log work](#)

On this page

- [Purpose](#)
- [Example: Add percentaged profit margin on issue closing](#)
- [Usage Examples](#)
- [Related Features](#)

Purpose







This post-function allows create automatic work logs when triggering transitions. Math expressions as complex as needed can be used to calculate the time to be logged, and custom parsed texts can be used for the comment.

SINCE VERSION 2.4.8

In version 2.4.8 the ability to [select a target issue](#), where the time should be logged, has been introduced.

Example: Add percentaged profit margin on issue closing

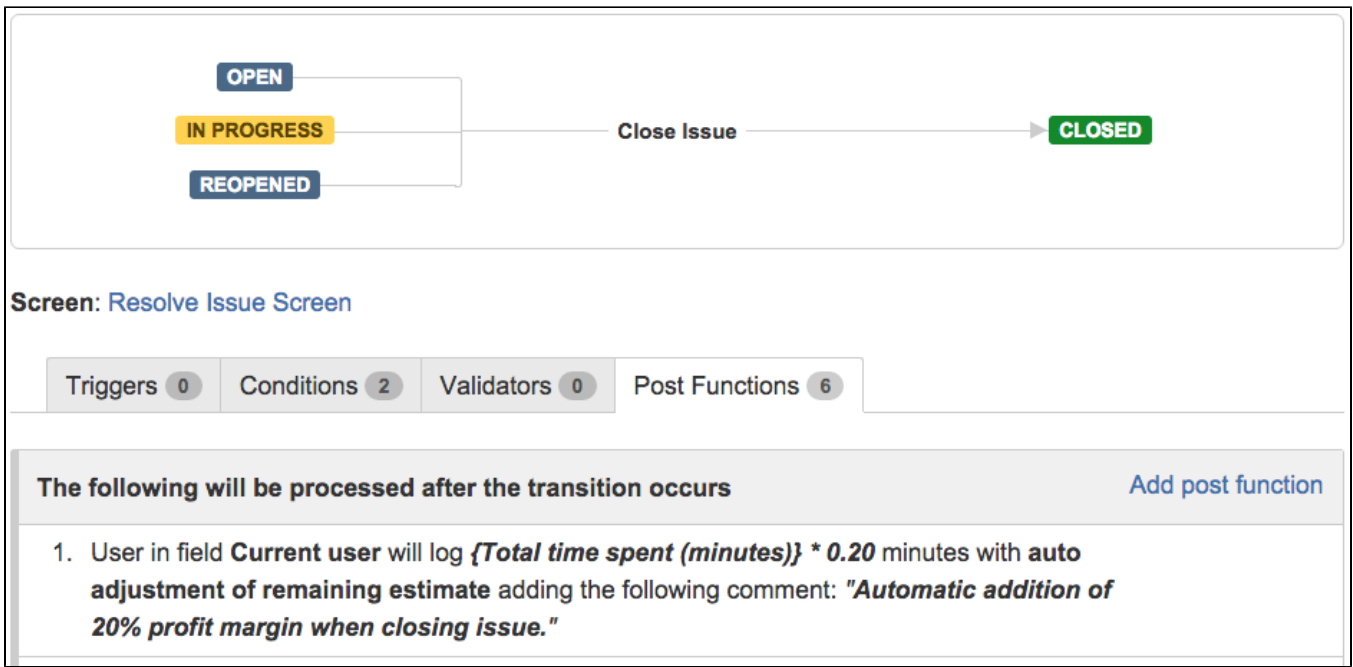
This configuration example shows how to add an automatic 20% profit margin in your issues when closing them:

Worklog author:	Assignee - [User] 	
Remaining estimate adjustment:	<input checked="" type="radio"/> Auto adjustment <input type="radio"/> Retain remaining estimate	
Time to be logged (minutes):	<div> <div>1</div> <div>{00025} * 0.2</div> </div> <div> Syntax Specification </div> <div> <div>Numeric/Date-Time field code injector:</div> <div>Original estimate (minutes) - [Number] - {00068} </div> <div> <p>Formula expressing a value in minutes. You can insert fields of type <i>Number</i>, <i>Date</i> and <i>Date-Time</i> using format {nnnnn}. Fields of type <i>Date</i> and <i>Date-Time</i> 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 ({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.</p> </div> <div>Check Syntax</div> </div>	
Starting Date-Time:	<div> <div>1</div> <div></div> </div> <div> Syntax Specification </div> <div> <div>Numeric/Date-Time field code injector:</div> <div>Original estimate (minutes) - [Number] - {00068} </div> <div> <p>Time expression for the starting date-time of the work log entry. In case no expression is entered, current date-time is used. You can insert fields of type <i>Number</i>, <i>Date</i> and <i>Date-Time</i> using format {nnnnn}. Fields of type <i>Date</i> and <i>Date-Time</i> represent milliseconds elapsed since January 1, 1970, 00:00:00 GMT.</p> </div> <div>Check Syntax</div> </div>	
Log comment:	<div> <div>1</div> <div>Automatic addition of 20% profit maging when closing the issue.</div> </div> <div> <div>Field code injector:</div> <div>Summary - [Text] - %{00000} </div> <div> <p>- Compose free text by inserting field codes %{nnnnn} that will be replaced by corresponding field values prior to be copied to target field.</p> <p>- You can insert parent and child values of cascading select fields writing %{nnnnn.0} for parent value, and %{nnnnn.1} for child value.</p> <p>- You can also insert values of multi-level cascading select fields writing %{nnnnn.1}, with 1 being the level to be read (root level is 0).</p> </div> <div>[Line 1 / Col 84]</div> </div>	
Conditional execution:	<div> <div>1</div> <div></div> </div> <div> <div>Optional boolean expression that should be satisfied in order to actually execute the post-function.</div> <div>(Syntax Specification)</div> </div> <div> <div>Leave the field empty for executing the post-function unconditionally.</div> <div>Collection of Examples</div> <div>[Line 1 / Col 1]</div> </div> <div> <div> <p><u>Logical connectives</u>: and, or and not. Alternatively you can also use &, and !.</p> <p><u>Comparison operators</u>: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !=~ can be used with <i>strings</i>, <i>multi-valued fields</i> and <i>lists</i>.</p> <p><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.</p> </div> <div>Check Syntax</div> </div> <div> <div> <div><u>String Field Code Injector:</u></div> <div>Summary - [Text] - %{00000} </div> </div> <div> <div><u>Numeric/Date Field Code Injector:</u></div> <div>Original estimate (minutes) - [Number] - {00068} </div> </div> </div>	

Note that:

- {00025} is field code for "Total time spent (minutes)"

Once configured, your "Close Issue" transition looks like this:



Usage Examples

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 Features