Limit the number of hours a user can log per day

On this page

Features used to implement the example

- **Boolean Validator with math, date-time or text-string terms**
- **Set a field as a function of other fields**
- **Set or create a user property**
- **Copy parsed text to a field**

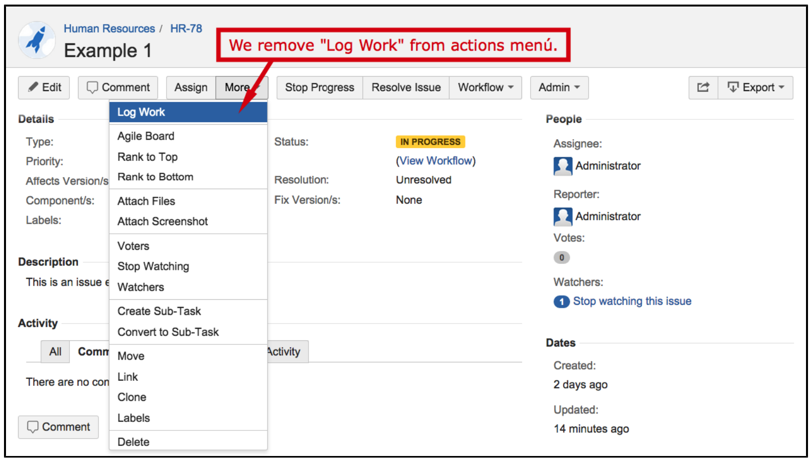
Example: Limit the number of hours a user can log per day

The following configuration allows to apply a restriction on the sum of hours a user can log in a single day, counting work logs in any issue.

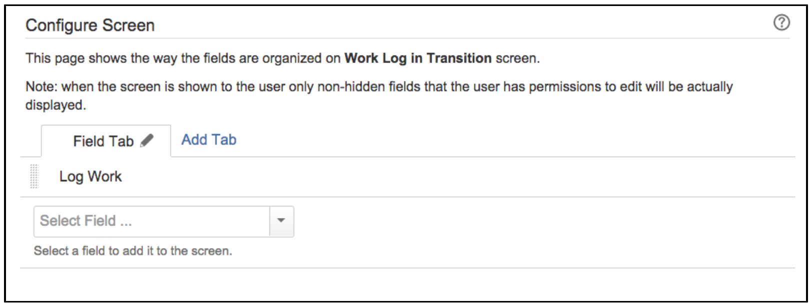
Enforce work logs through transitions

First we need to enforce users to log all the work using transitions in our workflows, instead of using "**Log Work**" operation at the issue screen. To do it we will:

1. Remove "**Log Work**" from actions menu: `Administration > Add-ons > Manage add-ons > Filter by "System" > Issue Operations Plugin > Disable module "View Issue Ops Bar Work Link"`



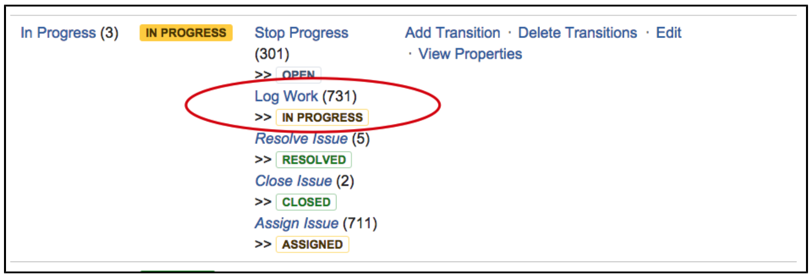
2. Create a screen with only one field: "**Log Work**". You can call it "**Work Log in Transition**".


Configure Screen

This page shows the way the fields are organized on **Work Log in Transition** screen.

Note: when the screen is shown to the user only non-hidden fields that the user has permissions to edit will be actually displayed.

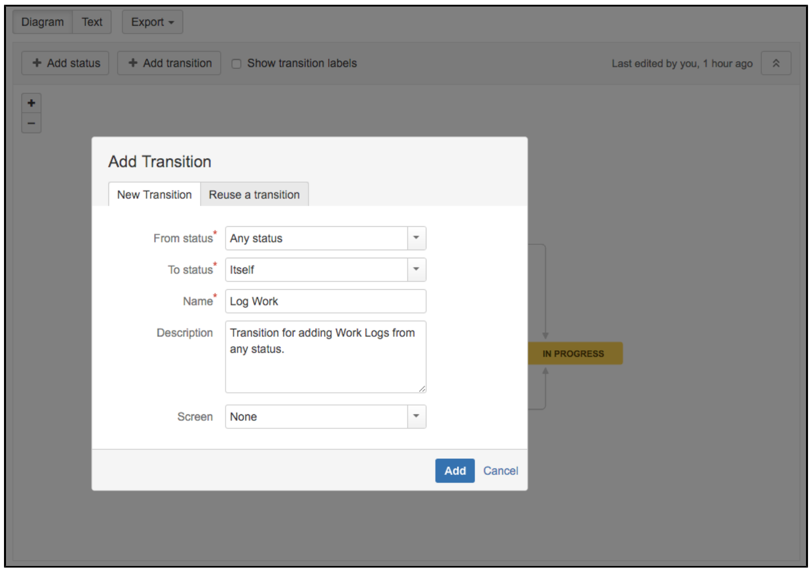
3. Add reflexive transition called **"Log Work"**: in the statuses of our workflow where we want users to be allowed to log work, and we associate them the screen "**Work Log in Transition**", created in the previous step. These transitions have the same status as origin and destination, leaving the issue in the same status, but showing the user a screen where they will be able to log work.


In Progress (3) IN PROGRESS

Stop Progress (301)
>> OPEN
Log Work (731)
>> IN PROGRESS
Resolve Issue (5)
>> RESOLVED
Close Issue (2)
>> CLOSED
Assign Issue (711)
>> ASSIGNED

Add Transition · Delete Transitions · Edit · View Properties

Typically you will add this transition to "**In Progress**" status, but you can add this transition to all statuses easily using a global reflexive transition.



Add Transition

New Transition | Reuse a transition

From status*: Any status
To status*: Itself
Name*: Log Work
Description: Transition for adding Work Logs from any status.
Screen: None

You can use conditions or validations to limit who can execute these transition, and thus who can log work.

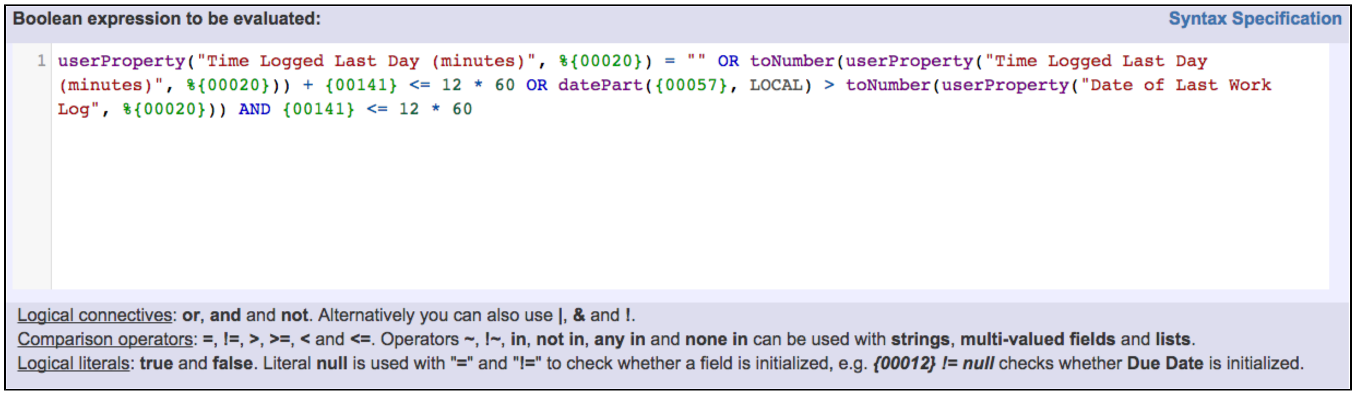
User properties

We will use **two user properties** to store in order to implement this restriction. These two user properties are **created and updated automatically**, so **we don't need to create them manually**:

- **Date of Last Work Log**: contains the date of the last work log done by the user. The date is stored as a number of milliseconds since fixed date.
- **Time Logged Last Day (minutes)**: contains the number of minutes logged by the user in the last day he has done any work logging.

Now we are ready to add validations and post-functions to transitions "**Log Work**", that will implement the restriction on the number of hours a user can log per day. In this particular example **we will restrict the sum of hours a user can log in any issue to 12 per day**:

Add **Boolean Validator with math, date-time or text-string terms** to transitions "**Log Work**" with the following configuration:


Boolean expression to be evaluated:

```
userProperty("Time Logged Last Day (minutes)", %{00020}) = "" OR toNumber(userProperty("Time Logged Last Day
(minutes)", %{00020})) + {00141} <= 12 * 60 OR datePart({00057}, LOCAL) > toNumber(userProperty("Date of Last Work
Log", %{00020})) AND {00141} <= 12 * 60
```

Logical connectives: **or**, **and** and **not**. Alternatively you can also use **|**, **&** and **!**.
Comparison operators: **=**, **!=**, **>**, **>=**, **<** and **<=**. Operators **~**, **!~**, **in**, **not in**, **any in** and **none in** 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.

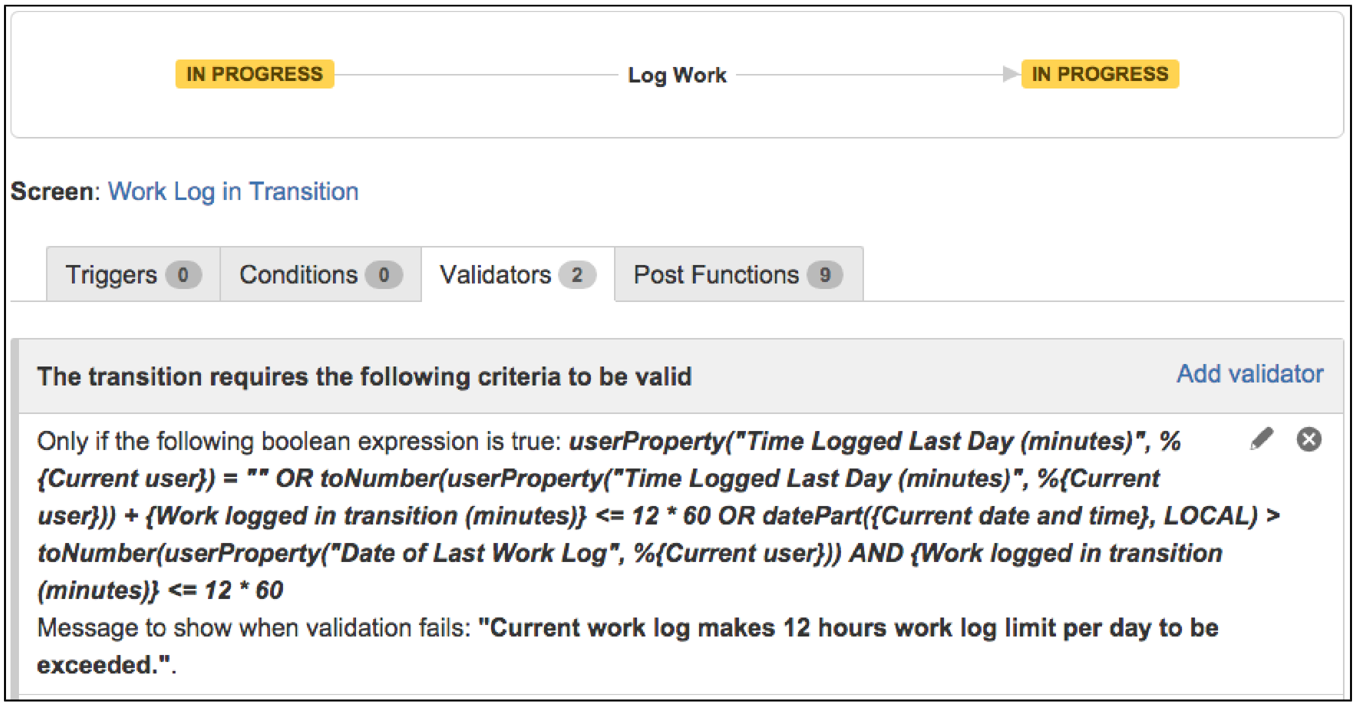
Boolean expression used is:

```
userProperty("Time Logged Last Day (minutes)", %{00020}) = "" OR toNumber(userProperty("Time Logged Last Day
(minutes)", %{00020})) + {00141} <= 12 * 60 OR datePart({00057}, LOCAL) > toNumber(userProperty("Date of Last
Work Log", %{00020})) AND {00141} <= 12 * 60
```

Note that:

- **%{00020}** is field code for virtual field "**Current user**"
- **{00057}** is code for numeric value of virtual field "**Current date and time**"
- **{00141}** is code for numeric value of virtual field "**Work logged in transition (minutes)**"

Once validation is added, transition "**Log Work**" will look like this:



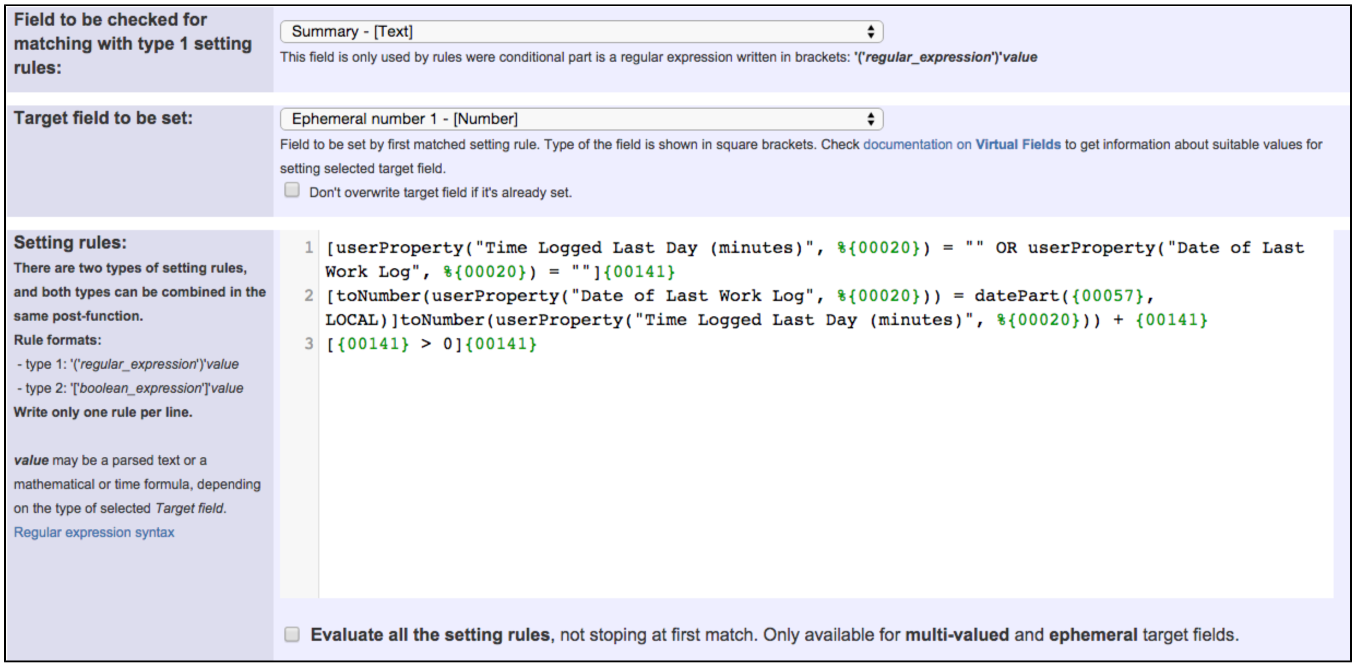
The transition requires the following criteria to be valid

Only if the following boolean expression is true: ***userProperty("Time Logged Last Day (minutes)", %{Current user}) = "" OR toNumber(userProperty("Time Logged Last Day (minutes)", %{Current user})) + {Work logged in transition (minutes)} <= 12 * 60 OR datePart({Current date and time}, LOCAL) > toNumber(userProperty("Date of Last Work Log", %{Current user})) AND {Work logged in transition (minutes)} <= 12 * 60***
Message to show when validation fails: **"Current work log makes 12 hours work log limit per day to be exceeded."**.

We will add 4 post-functions to transitions "**Log Work**". Now we describe each of them by **execution order**:

Post-function 1: **Set a field as a function of other fields**

This post-function will store in "**Ephemeral number 1**" the value that should take user property "**Time Logged Last Day (minutes)**" after transition execution:


Field to be checked for matching with type 1 setting rules: Summary - [Text]
This field is only used by rules were conditional part is a regular expression written in brackets: '('*regular_expression*')'*value*

Target field to be set: Ephemeral number 1 - [Number]
Field to be set by first matched setting rule. Type of the field is shown in square brackets. Check documentation on **Virtual Fields** to get information about suitable values for setting selected target field.
☐ Don't overwrite target field if it's already set.

Setting rules:
There are two types of setting rules, and both types can be combined in the same post-function.
Rule formats:
- type 1: '('*regular_expression*')'*value*
- type 2: '['*boolean_expression*']'*value*
Write only one rule per line.

value may be a parsed text or a mathematical or time formula, depending on the type of selected *Target field*.
Regular expression syntax

```
[userProperty("Time Logged Last Day (minutes)", %{00020}) = "" OR userProperty("Date of Last Work Log", %{00020}) = ""]{00141}
[toNumber(userProperty("Date of Last Work Log", %{00020})) = datePart({00057}, LOCAL)]toNumber(userProperty("Time Logged Last Day (minutes)", %{00020})) + {00141}
[{00141} > 0]{00141}
```

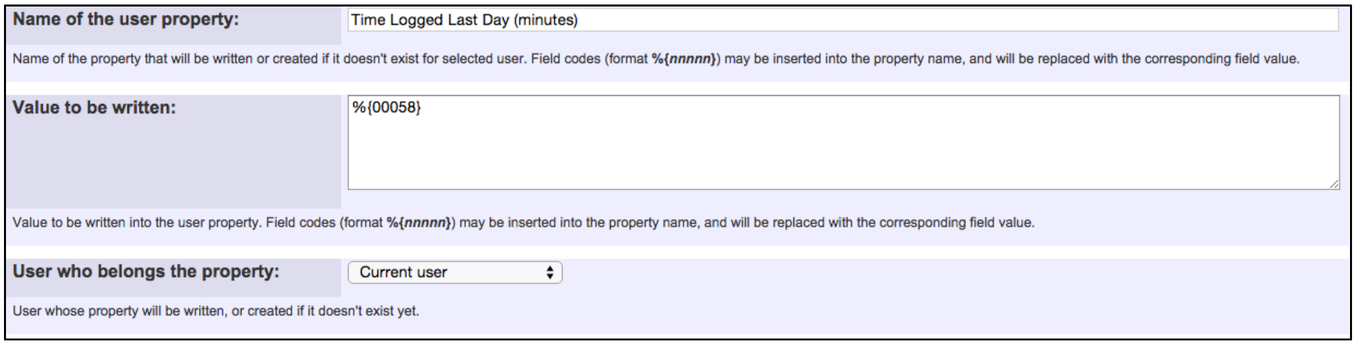
☐ **Evaluate all the setting rules**, not stoping at first match. Only available for **multi-valued** and **ephemeral** target fields.

The 3 setting rules (one rule per line) used are:

1. **`[userProperty("Time Logged Last Day (minutes)", %{00020}) = "" OR userProperty("Date of Last Work Log", %{00020}) = ""]{00141}`**
2. **`[toNumber(userProperty("Date of Last Work Log", %{00020})) = datePart({00057}, LOCAL)]toNumber(userProperty("Time Logged Last Day (minutes)", %{00020})) + {00141}`**
3. **`[{00141} > 0]{00141}`**

Post-function 2: **Set or create a user property**

This post-function will set (or create in case it does not exist yet) user property "**Time Logged Last Day (minutes)**" with the value stored in "**Ephemeral number 1**":


Name of the user property: Time Logged Last Day (minutes)
Name of the property that will be written or created if it doesn't exist for selected user. Field codes (format **%{*nnnnn*}**) may be inserted into the property name, and will be replaced with the corresponding field value.

Value to be written: %{00058}
Value to be written into the user property. Field codes (format **%{*nnnnn*}**) may be inserted into the property name, and will be replaced with the corresponding field value.

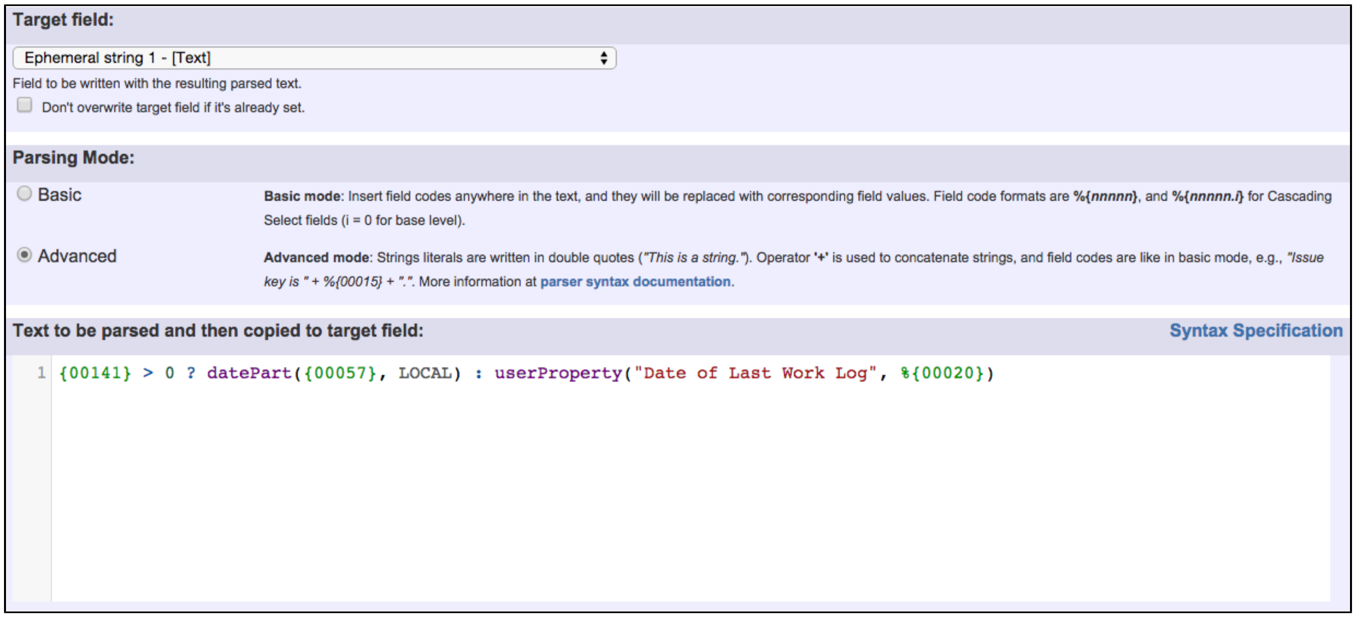
User who belongs the property: Current user
User whose property will be written, or created if it doesn't exist yet.

Note that:

- **%{00058}** is field code for virtual field "**Ephemeral number 1**"

Post-function 3: Copy parsed text to a field

This post-function will store in "**Ephemeral string 1**" the value that should take user property "**Date of Last Work Log**" after transition execution:


Target field:

Ephemeral string 1 - [Text]

Field to be written with the resulting parsed text.

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

Parsing Mode:

○ Basic — **Basic mode**: Insert field codes anywhere in the text, and they will be replaced with corresponding field values. Field code formats are **%{nnnnn}**, and **%{nnnnn.i}** for Cascading Select fields (i = 0 for base level).

● Advanced — **Advanced mode**: Strings literals are written in double quotes (*"This is a string."*). Operator **'+'** is used to concatenate strings, and field codes are like in basic mode, e.g., *"Issue key is " + %{00015} + "."*. More information at parser syntax documentation.

Text to be parsed and then copied to target field: Syntax Specification

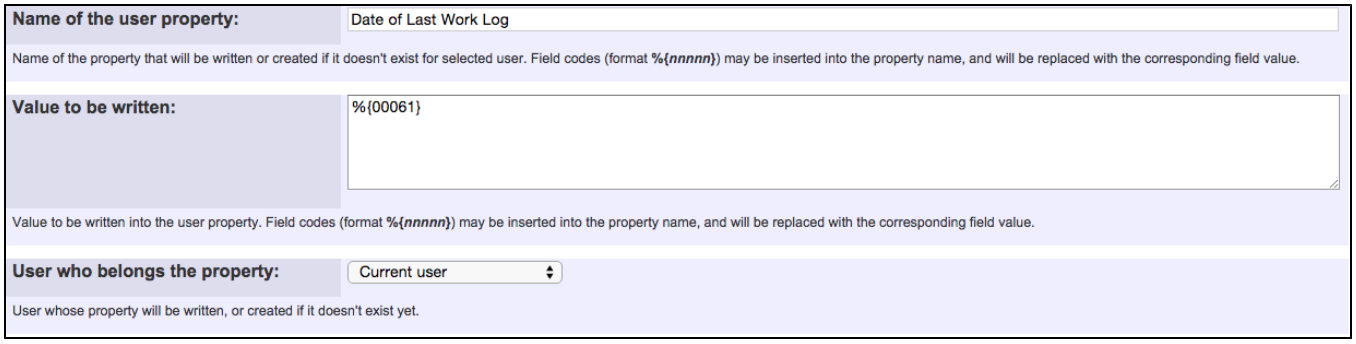
```
{00141} > 0 ? datePart({00057}, LOCAL) : userProperty("Date of Last Work Log", %{00020})
```

Text to be parsed in **advanced mode** is:

`{00141} > 0 ? datePart({00057}, LOCAL) : userProperty("Date of Last Work Log", %{00020})`

Post-function 4: Set or create a user property

This post-function will set (or create in case it does not exist yet) user property "**Date of Last Work Log**" with the value stored in "**Ephemeral string 1**":


Name of the user property: Date of Last Work Log

Name of the property that will be written or created if it doesn't exist for selected user. Field codes (format **%{nnnnn}**) may be inserted into the property name, and will be replaced with the corresponding field value.

Value to be written: %{00061}

Value to be written into the user property. Field codes (format **%{nnnnn}**) may be inserted into the property name, and will be replaced with the corresponding field value.

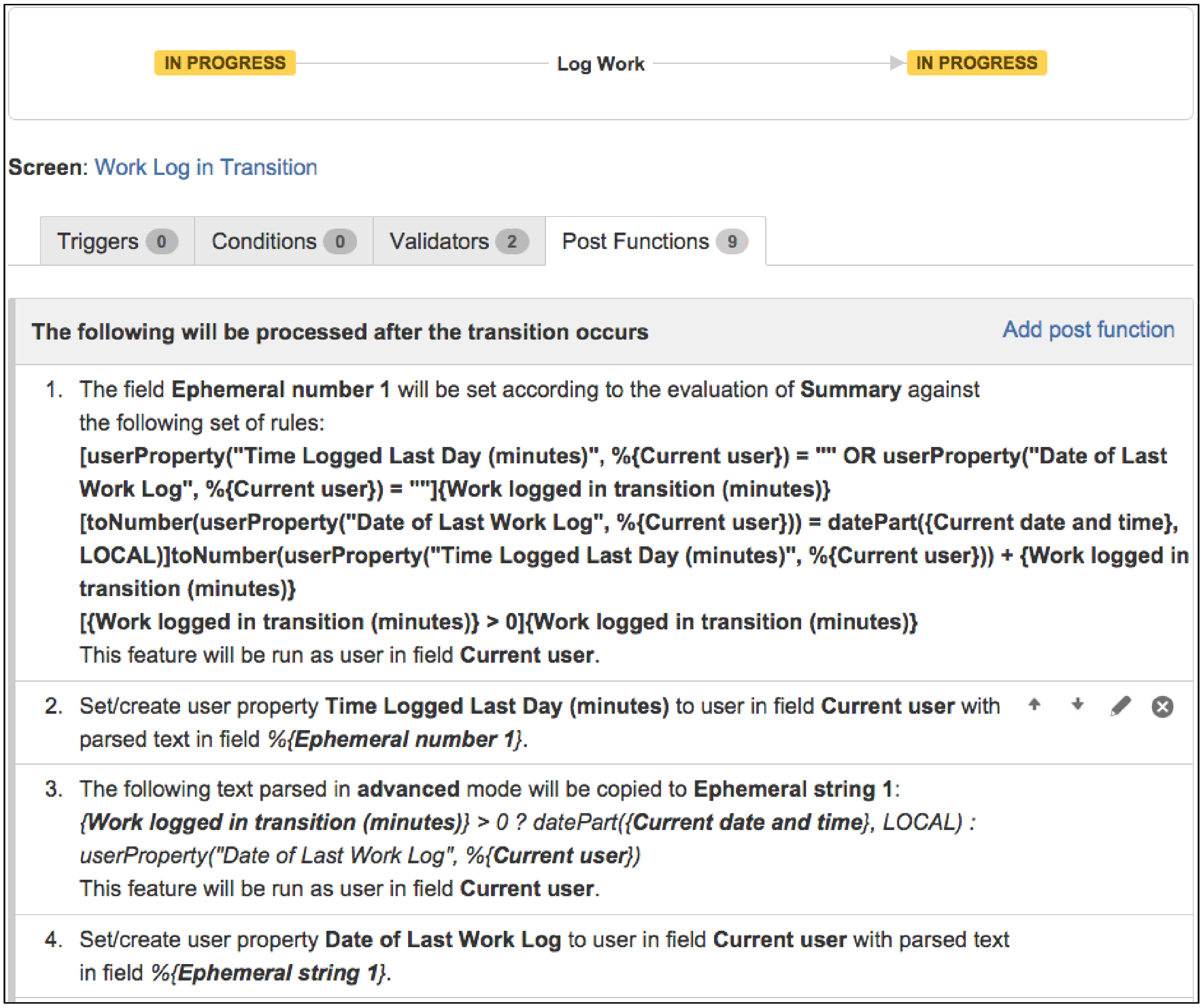
User who belongs the property: Current user

User whose property will be written, or created if it doesn't exist yet.

Note that:

- **%{00061}** is field code for virtual field "**Ephemeral string 1**"

Once all the post-functions have been inserted, transition "**Log Work**" will look like this:



Screen: Work Log in Transition

The following will be processed after the transition occurs

1. The field **Ephemeral number 1** will be set according to the evaluation of **Summary** against the following set of rules:
   [userProperty("Time Logged Last Day (minutes)", %{Current user}) = "" OR userProperty("Date of Last Work Log", %{Current user}) = ""]{Work logged in transition (minutes)}
   [toNumber(userProperty("Date of Last Work Log", %{Current user})) = datePart({Current date and time}, LOCAL)]toNumber(userProperty("Time Logged Last Day (minutes)", %{Current user})) + {Work logged in transition (minutes)}
   [{Work logged in transition (minutes)} > 0]{Work logged in transition (minutes)}
   This feature will be run as user in field **Current user**.
2. Set/create user property **Time Logged Last Day (minutes)** to user in field **Current user** with parsed text in field *%{**Ephemeral number 1**}*.
3. The following text parsed in **advanced** mode will be copied to **Ephemeral string 1**:
   *{**Work logged in transition (minutes)**} > 0 ? datePart({**Current date and time**}, LOCAL) : userProperty("Date of Last Work Log", %{**Current user**})*
   This feature will be run as user in field **Current user**.
4. Set/create user property **Date of Last Work Log** to user in field **Current user** with parsed text in field *%{**Ephemeral string 1**}*.

Other examples of that functions

Boolean Validator with math, date-time or text-string terms

Page: Block a transition until all sub-tasks have certains fields populated
Page: Block an epic's transition depending on linked issues status and due date
Page: Block or hide a transition for an issue depending on its issue links
Page: Block or unblock a transition after an issue rested a specific time in a status
Page: Block transition until all sub-tasks are in a specific status category
Page: Close parent issue when all sub-tasks are closed
Page: Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
Page: Ensure that all issues linked with a certain issue link type have "Due Date" field set
Page: If field A is populated then, field B must also be populated
Page: Limit issue creation per role and issue type
Page: Limit the number of hours a user can log per day
Page: Limit valid dates for work logs
Page: Make "Time Spent" field required when there is no time logged in the issue
Page: Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"

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
- 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

Page: Make attachment mandatory depending on the value of certain custom field
Page: Make different fields mandatory depending on the value of a Select List custom field
Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows
Page: Make parent issue progress through its workflow
Page: Prevent issue creation if another issue with same field value already exists
Page: Reject duplicated file names in attachments
Page: Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
Page: Require issue link when resolving as duplicate
Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
Page: Restrict sub-task type creation depending on parent issue status
Page: Restrict sub-task type creation depending on parent issue type
Page: Set a condition in a global transition which only applies in a certain status
Page: Validate a custom field "Story Points" has been given a value in Fibonacci sequence
Page: Validate compatible values selection among dependent custom fields
Page: Validate only issue links created in transition screen
Page: Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B
Page: Validation and condition based on time expressions
Page: Validation based on the value of a date type project property
Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments
Page: Validation on sibling sub-tasks depending on issue type and status
Page: Validation on the value of a Cascading Select field

Set a field as a function of other fields

Page: Add watcher depending on security level
Page: Add watchers based on issue type
Page: Add watchers depending on the value of a custom field
Page: Assign issue based on the value of a Cascading Select custom field
Page: Assign issue to a specific user based on a specific custom field value
Page: Assign issue to current user if assignee is empty
Page: Assign issue to current user if the user is not member of a certain project role
Page: Change assignee based on a custom field
Page: Change parent's status depending on sub-task's summary
Page: Changing issue priority depending on issue description
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: Limit the number of hours a user can log per day
Page: Make parent issue progress through its workflow
Page: Rise priority if due date is less than 3 weeks away
Page: Set "Due date" depending on the value of other fields, in case it's uninitialized
Page: Set "Due date" to a specific day of next week no matter of date of creation this week
Page: Set "Due date" to current date at issue creation if not initialized
Page: Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field
Page: Set a date based on current date
Page: Set a field based on reporter's email
Page: Set a watcher at ticket creation depending on custom field's value
Page: Set assignee depending on issue type
Page: Set security level based on groups and project roles the reporter or creator are in
Page: Set security level depending on reporter or creator
Page: Set the assignee based on a condition
Page: Set the value of a field of type "User Picker" depending on other field's value
Page: Set watchers depending on the value of a custom field
Page: Setting a custom field (User Picker) based on the value of another custom field (Text Field)
Page: Setting a field's default value depending on another field
Page: Setting the priority depending on the multiplication of custom fields
Page: Transition an issue automatically depending on the value of a field
Page: Unassign an issue when assigned to project leader

- 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
- 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

Page: Update checkboxes custom field if a file has been attached during a transition
Page: Using project properties to calculate custom sequence numbers

---

Set or create a user property

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: Add current user to comment
Page: Add or remove request participants
Page: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."
Page: Assign issue based on the value of a Cascading Select custom field
Page: Assign issue to last user who executed a certain transition in the workflow
Page: Automatically close resolved sub-tasks when parent issue is closed
Page: Automatically reopen parent issue when one of its sub-tasks is reopened
Page: Calculate the time elapsed between 2 transition executions
Page: Close parent issue when all sub-tasks are closed
Page: Combine the values of several Multi-User picker fields
Page: Compose a parsed text including the "full name" or a user selected in a User Picker custom field
Page: Compose dynamic text by inserting field values in a text template
Page: Copy issue labels to a custom field
Page: Copy the value of a user property into a user picker
Page: Create a comment in sub-tasks when parent transitions
Page: Execute transition in epic
Page: Getting the number of selected values in a custom field of type Multi Select
Page: Limit the number of hours a user can log per day
Page: Make a sub-task's status match parent issue's current status on creation
Page: Make parent issue progress through its workflow
Page: Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"
Page: Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
Page: Parse Email adresses to watchers list
Page: Parsing text from last comment and appending it to issue's summary
Page: Remove versions selected in a version picker custom field
Page: Replace certain issue link types with different ones
Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
Page: Set a Select or Multi-Select field using regular expression to express the values to be assigned
Page: Set assignee depending on issue type
Page: Set field depending on time passed since issue creation
Page: Set priority for issues that have been in a certain status for longer than 24 hours
Page: Set security level based on groups and project roles the reporter or creator are in
Page: Transition linked issues in currently active sprint
Page: Transition only a sub-task among several ones
Page: Transition parent issue only when certain issue sub-task types are done
Page: Update Cascading Select custom field with a value of the field in parent issue
Page: Update checkboxes custom field if a file has been attached during a transition
Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments
Page: Writing a comment to blocked issues when blocking issues are resolved

---

Copy parsed text to a field

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: Add current user to comment
Page: Add or remove request participants

Page: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."
Page: Assign issue based on the value of a Cascading Select custom field
Page: Assign issue to last user who executed a certain transition in the workflow
Page: Automatically close resolved sub-tasks when parent issue is closed
Page: Automatically reopen parent issue when one of its sub-tasks is reopened
Page: Calculate the time elapsed between 2 transition executions
Page: Close parent issue when all sub-tasks are closed
Page: Combine the values of several Multi-User picker fields
Page: Compose a parsed text including the "full name" or a user selected in a User Picker custom field
Page: Compose dynamic text by inserting field values in a text template
Page: Copy issue labels to a custom field
Page: Copy the value of a user property into a user picker
Page: Create a comment in sub-tasks when parent transitions
Page: Execute transition in epic
Page: Getting the number of selected values in a custom field of type Multi Select
Page: Limit the number of hours a user can log per day
Page: Make a sub-task's status match parent issue's current status on creation
Page: Make parent issue progress through its workflow
Page: Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"
Page: Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
Page: Parse Email adresses to watchers list
Page: Parsing text from last comment and appending it to issue's summary
Page: Remove versions selected in a version picker custom field
Page: Replace certain issue link types with different ones
Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
Page: Set a Select or Multi-Select field using regular expression to express the values to be assigned
Page: Set assignee depending on issue type
Page: Set field depending on time passed since issue creation
Page: Set priority for issues that have been in a certain status for longer than 24 hours
Page: Set security level based on groups and project roles the reporter or creator are in
Page: Transition linked issues in currently active sprint
Page: Transition only a sub-task among several ones
Page: Transition parent issue only when certain issue sub-task types are done
Page: Update Cascading Select custom field with a value of the field in parent issue
Page: Update checkboxes custom field if a file has been attached during a transition
Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments
Page: Writing a comment to blocked issues when blocking issues are resolved