

Copy the value of a user property into a user picker

On this page

- [Features used to implement the example](#)
- [Example: Copy the value of a user property into a user picker](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Read user property](#)
- [Copy parsed text to a field](#)

Example: Copy the value of a user property into a user picker

We are using your [Jira Workflow Toolbox](#) plugin, and would like the ability to use the post-function [Read user property](#) to copy a value into a user picker field.. For example we have the user's manager stored as a user property, and we would like to copy that into a custom user picker field for use in certain workflows.

You have to use post-function [Read user property](#). This post-function allows you to read a user property, but doesn't let you copy the value directly into a user picker. As a workaround you can use auxiliary field "**Ephemeral string 1**" as temporary storage, and then post-function [Copy parsed text to a field](#) to copy its value into the desired user picker custom field. Next version of the plugin (2.0.25) will enable post-function [Read user property](#) to copy the value directly into a user picker among other many types of custom fields and virtual fields.

We use post-function [Read user property](#) to copy the value of user property "**Manager**" in current user (logged user) into virtual field "**Ephemeral string 1**":

Name of property to be read:	Manager
You can insert field codes into the user property name that will be replaced with the corresponding field value.	
Field to hold property value:	Ephemeral string 1
Value of user property will be written into the selected field.	
Property owner:	Current user

We use post-function [Copy parsed text to a field](#) the value in "**Ephemeral string 1**" into a user picker custom field "**Authorizer**":

Target field:

Authorizer

Text to be parsed and then copied to target field:

%{00061}

- **Compose a free text** by inserting field codes that will be replaced by corresponding field values prior to be copied to target field.
- You can reference parent and child values of **cascading select fields** writing `%{xxxxx.0}` for parent value, and `%{xxxxx.1}` for child value.
- You can change **reporter, assignee, due date, issue status, priority, resolution, labels, components, fixed versions, affected versions, original estimate, estimated, time spent, and security level** by choosing the suitable target field and value to be assigned.
- To assign **cascading selects, multi selects, multi checkboxes, components, labels, fixed versions and affected versions** you should use comma or semicolon separated values.
- Additionally, fields of type **Select list, Radio button, Multi select, Multicheck box, Multi user, Multi groups, Components and Versions** can be set through regular expressions: options that matches a regular expression can be set by writing `//(regular_expression)/`, and options that doesn't match a regular expression can be set by writing `//(regular_expression)/`.
- **Setting and unsetting individual values** in multi-valued fields, leaving the rest untouched, can be achieved simply by inserting a character '+' or '-' preceding the value or the list of values. You can insert more than one '+' / '-' character in a sole setting operation.
- Fields **Attachments (only new attachments will be added)** and **Attachments (all current attachments will be replaced)** expect one or more issue keys whose attachments will be copied to current issue.
- You can also use this post-function to **cast a string into a number**.

Once configured, the transition looks like this:

All
Conditions (3)
Validators (1)
Post Functions (7)

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

Read user property **Manager** from user in **Current user** an write it's value into field **Ephemeral string 1**.

Edit | Move Down | Delete

THEN

The following parsed text will be copied to **Authorizer**:

#Ephemeral string 1#

Edit | Move Up | Move Down | Delete

Other examples of that function

Page: [Copy the value of a user property into a user picker](#)

Related Usage Examples

- [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
- Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
 - example
 - validator
 - custom-field
- Make attachment mandatory depending on the value of certain custom field
 - example
 - validator
 - custom-field
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
 - example
 - post-function
 - custom-field
 - sub-task
- Total of all story points in an epic
 - example
 - custom-field
 - calculated-field
- Show timeliness of an issue based on two date pickers
 - example
 - custom-field
 - calculated-field
- Add and remove a single or a set of items from multi valued fields
 - example
 - post-function
 - custom-field
 - issue-links
 - sub-task
- Highest value of a custom field among linked issues
 - example
 - custom-field
 - calculated-field
- Google Maps location from address
 - example
 - calculated-field
 - custom-field
- Make certain custom field required in resolve screen only if the resolution was set to "Fixed"
 - example
 - validator
 - custom-field