

# Assign issue based on the value of a Cascading Select custom field

## On this page

- [Features used to implement the example](#)
- [Example: Assign issue based on the value of a Cascading Select custom field](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

## Features used to implement the example

- [Set a field as a function of other fields](#)
- [Copy parsed text to a field](#)

## Example: Assign issue based on the value of a Cascading Select custom field

We have bought [Jira Workflow Toolbox](#) and I am trying to develop a ticketing system which will assign a ticket based on the results of the cascading select that user will choose. Which post function can I use from your plugin?

To do it you have to use two post-functions:

- We use post-function [Copy parsed text to a field](#) to copy into auxiliary virtual field "**Ephemeral string 1**" the value of one of the two levels of the cascading select field (let's call them Level 0 and Level 1).
- We use post-function [Set a field as a function of other fields](#) to copy into virtual field "**Assignee**" the name of a user or the name of a project role, depending on the value of virtual field "**Ephemeral string 1**". If you are going to use project roles to assign the issue, you should have previously [configured the default user for those project roles in your projects](#).

Let's suppose we have a custom field of type cascading select called "**Country**". In Level 0 we select a continent, and in Level 1 we select a country in previously selected continent. Options available in custom field "**Country**":

Applicable contexts for scheme: [Edit Configuration](#)

Issue type(s):  
Global (all issues)

Default Value: [Edit Default Value](#)

Options: [Edit Options](#)

- **Europe**  
Spain, France, United Kingdom, Germany, Italy
- **Asia**  
Japan, China, India
- **Africa**  
Egypt, Kenya, Morocco, South Africa
- **America**  
USA, Canada, Mexico, Argentina, Brasil, Colombia
- **Oceania**  
New Zeland, Australia

We use post-function [Copy parsed text to a field](#) to copy Level 0 of field "Country" into auxiliary virtual field "Ephemeral string 1":

**Target field:**  

Ephemeral string 1

**Text to be parsed and then copied to target field:**  

%{10900.0}

- **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**.

Note the notation used to reference level 0 of field Country:

- `%{10900.0}`. In case you want to reference level 1 you should use `%{10900.1}`.
- `10900` is the ID of custom field "Country" in our example

We use post-function [Set a field as a function of other fields](#) to set the value of virtual field "Assignee" with the name of certain users of Jira, depending on the value of virtual field "Ephemeral string 1":

**Field to be checked for matching with the set of setting rules:**  

Ephemeral string 1

**Setting rules:**  
Put only one rule per line.  
Rule format: `"(regular_expression)"value`  
[Regular expression syntax](#)  

(Asia)aiguo  
(Europe)franz  
(Oceania)amy  
(America)dennis  
(Africa)kazemde

Write **only one rule per line**. The rules will be processed in order. Once a rule is matched by the field under evaluation, its associated value will be parsed and copied to selected target field, and the rest of the rules won't be processed.

If selected target field is of type **number, date or date and time**, the associated value should be a number or a mathematical/time formula. Other types like **user, date, issue status, issue priority and issue resolution** require values of corresponding suitable types.

Rule format: `"(regular_expression)"value`  
Both, regular expression and value will be parsed like in post-function "Copy parsed text to a field", this way, by inclusion of field codes, you will be able to create **dynamic regular expressions and assignable values**.

**Target field to be set:**  

Assignee

Once configured, the transition looks like this:

All Conditions (1) Validators (0) Post Functions (8)

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

The following parsed text will be copied to **Ephemeral string 1**:  
**#Country.0#**  
Edit | Move Down | Delete

**THEN**

The field **Assignee** will be set according to the evaluation of **Ephemeral string 1** against the following set of rules:  
(Asia)aiguo  
(Europe)franz  
(Oceania)amy  
(America)dennis  
(Africa)kazemde  
Edit | Move Up | Move Down | Delete

## Other examples of that functions

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

### Copy parsed text to a field

## Related Usage Examples

- [Make different fields mandatory depending on the value of a Select List custom field](#)
  - [example](#)
  - [validator](#)
  - [custom-field](#)
- [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 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](#)

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 addresses 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

- 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