# Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."

**On this page**

## Features used to implement the example

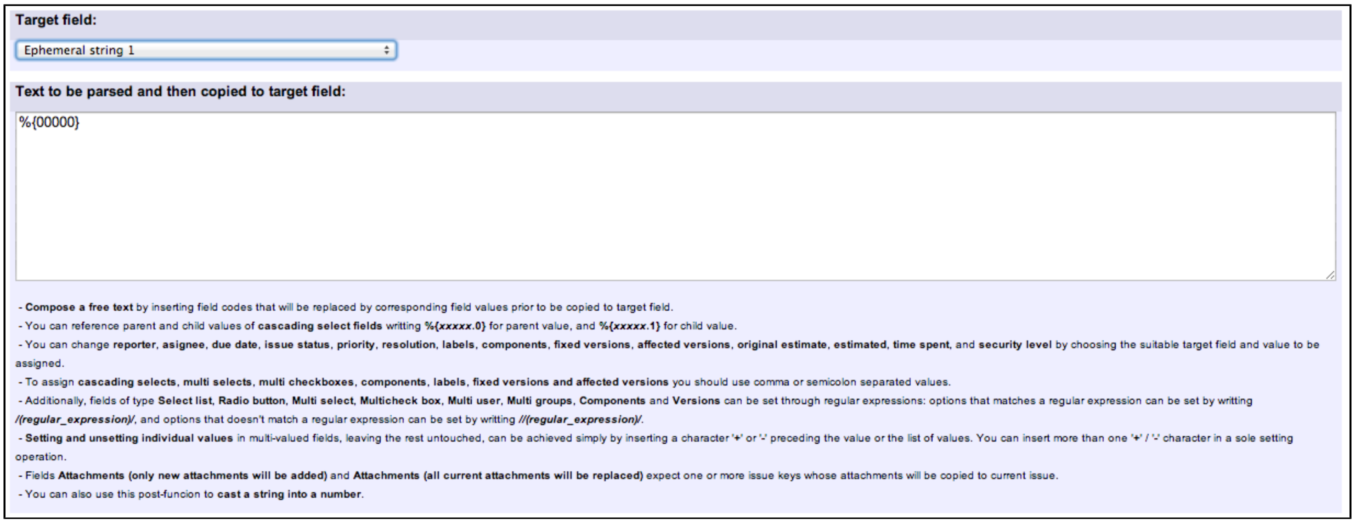
- **Copy parsed text to a field**
- **Regular expression renderer**
- **Add or remove watchers**

## Example: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."

As in my answer to your previous question you should move the post-function below "**Creates issue originally**".

On the other hand to add as watchers the list of comma separated users in summary, you should remove the previous part of the summary with post-function **Regular expression renderer**. I show you exactly how to do it:

Use post-function **Copy parsed text to a field** to copy "**Summary**" to an auxiliary field (**Ephemeral string 1**) that can be modified in second step by **Regular expression renderer**



**Target field:**

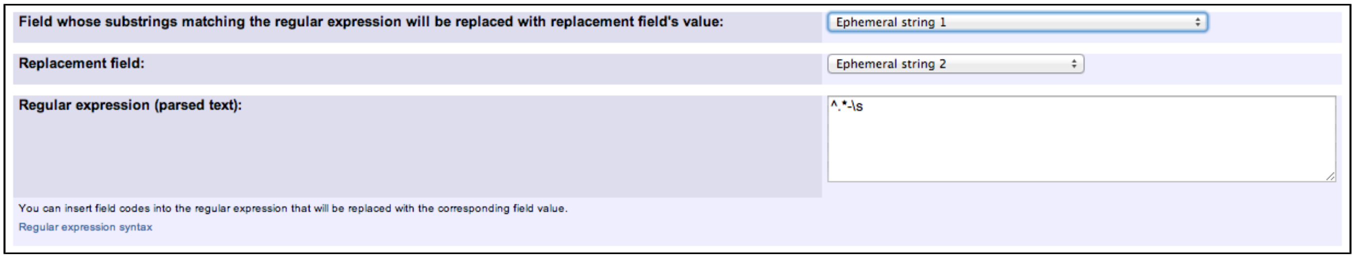
Ephemeral string 1

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

%{00000}

- **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** writting ***%{xxxxx.0}*** for parent value, and ***%{xxxxx.1}*** for child value.
- You can change **reporter, asignee, 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 writting ***/(regular_expression)/***, and options that doesn't match a regular expression can be set by writting ***//(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-funcion to **cast a string into a number**.

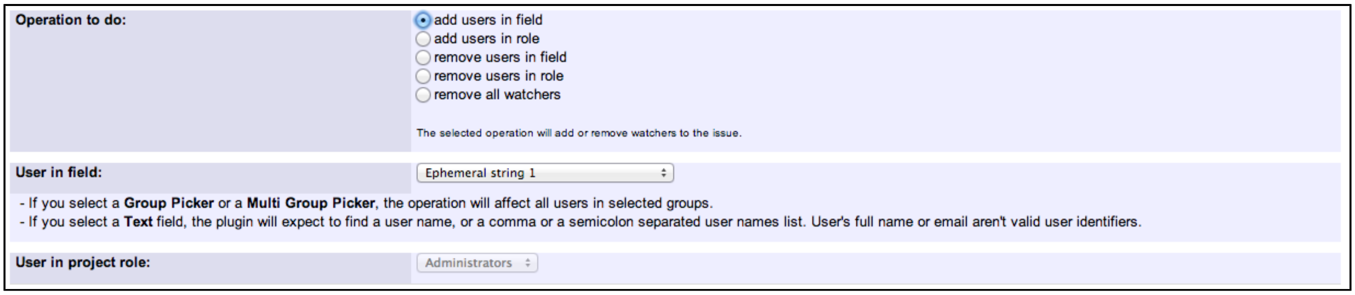
Use post-function **Regular expression renderer** to remove the part of the summary preceding the comma separated list of users. We use "**Ephemeral string 2**" as replacement string because it isn't initialized. Regular expression utilized is "**^.*-\s**"



| | |
|---|---|
| **Field whose substrings matching the regular expression will be replaced with replacement field's value:** | Ephemeral string 1 |
| **Replacement field:** | Ephemeral string 2 |
| **Regular expression (parsed text):** | ^.*-\s |

You can insert field codes into the regular expression that will be replaced with the corresponding field value.
Regular expression syntax

Use post-function **Add or remove watchers** to add users in "**Ephemeral string 1**" as issue watchers:

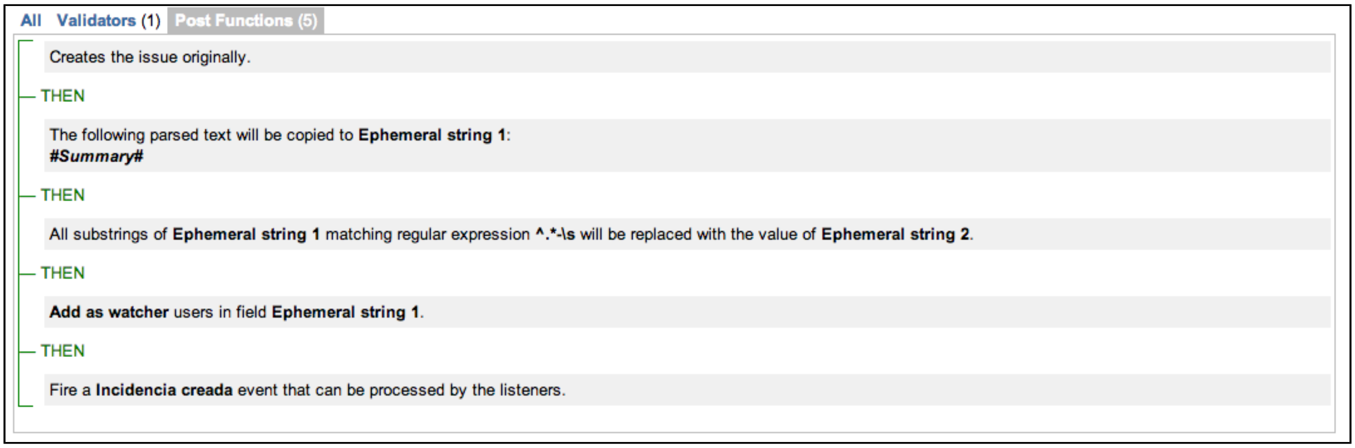


| | |
|---|---|
| **Operation to do:** | ● add users in field<br>○ add users in role<br>○ remove users in field<br>○ remove users in role<br>○ remove all watchers<br>The selected operation will add or remove watchers to the issue. |
| **User in field:** | Ephemeral string 1 |

- If you select a **Group Picker** or a **Multi Group Picker**, the operation will affect all users in selected groups.
- If you select a **Text** field, the plugin will expect to find a user name, or a comma or a semicolon separated user names list. User's full name or email aren't valid user identifiers.

| | |
|---|---|
| **User in project role:** | Administrators |

Once configured "**Create Issue**" transition looks like this:



Creates the issue originally.

THEN

The following parsed text will be copied to **Ephemeral string 1**:
***#Summary#***

THEN

All substrings of **Ephemeral string 1** matching regular expression **^.*-\s** will be replaced with the value of **Ephemeral string 2**.

THEN

**Add as watcher** users in field **Ephemeral string 1**.

THEN

Fire a **Incidencia creada** event that can be processed by the listeners.

# Other examples of that functions

### 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

# Related Usage Examples

- Creating a Jira Service Desk internal comment
  - example
  - post-function
- Limit the number of hours a user can log per day
  - example
  - validator
  - post-function
  - work-log
- Using project properties to calculate custom sequence numbers
  - example
  - post-function
  - calculated-field
  - project-properties
- Set a date based on current date
  - example
  - post-function

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

---

### Regular expression renderer

Page: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."

---

### Add or remove watchers

Page: Add watchers based on issue type
Page: Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."
Page: Set a watcher at ticket creation depending on custom field's value
Page: Set or remove watchers from a workflow transition
Page: Set watchers depending on the value of a custom field

- Setting the priority depending on the multiplication of custom fields
    - example
    - calculated-field
    - post-function
- Parse Email adresses to watchers list
    - example
    - post-function
- Set the assignee based on a condition
    - example
    - post-function
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
    - example
    - post-function
    - custom-field
    - sub-task
- Create a static set of sub-tasks with unique summaries
    - example
    - post-function
- Triage Jira Service Desk email requests (Move issues)
    - example
    - post-function
    - move
    - transition-issue
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)
    - example
    - post-function
    - transition
- Transition sub-tasks when parent is transitioned
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Transition only a sub-task among several ones
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
    - example
    - post-function
    - sub-task
    - transition
    - outdated