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

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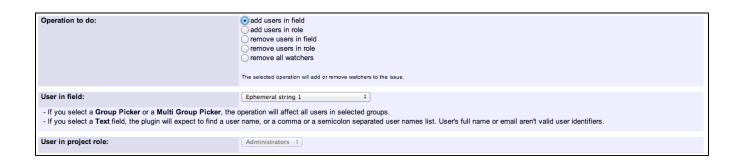
Use post-function **Copy parsed text to a field** to copy "**Summary**" to an auxiliary field (**Ephemeral string 1**) that can be modified in second step by **Regular expression renderer**

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** writing **%{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 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-funcion to **cast a string into a number**.
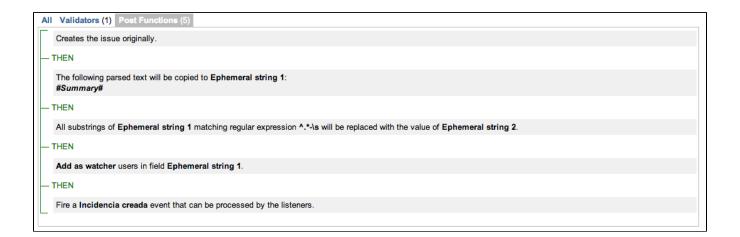

Use post-function **Regular expression renderer** to remove the part of the summary preceding the comma separated list of users. We use "**Ephemeral string 2**" as replacement string because it isn't initialized. Regular expression utilized is "**^.*-\s**"

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
	○ add users in role
	○ remove users in field
	○ remove users in role
	○ remove all watchers
	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:

All Validators (1) Post Functions (5)

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

Regular expression renderer

Add or remove watchers