

Parse Email adresses to watchers list

On this page

- [Features used to implement the example](#)
- [Example: Parse Email addresses to watchers list](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Update issue fields](#)

Example: Parse Email addresses to watchers list

In this example, a comma separated list of email addresses of Jira users are entered in a text custom field. By evaluating these addresses, the specific Jira users should be added to the watchers list on issue creation.

This can be done using the [Update issue fields](#) post function:

- Select the **New Watchers** virtual field and confirm by clicking **Add**
- Switch **Value type** to **Parsed text (advanced mode)**
- Add Text to be parsed: `toString(textOnStringList(toStringList(%{nnnnn}, ", "), toString(usersWithEmail(^%))))`

Update parameters of the Update issue fields Function for this transition.

Update parameters of the Update issue fields Function for this transition.

Target fields and Source values:

Select the target fields that will be set and the source values for each of them.

Target field:

Start typing to get the list of available fields

Add

Add a field to be set in selected issues.

Target Field	Type of Value	Source Value	Don't Overwrite	Actions
New watchers	Parsed text (advanced mode)	toString(textOnStringList(toStringList(%{CF Text Field (single line)}, ", "), toString(usersWithEmail(^%))))		Edit Remove

Note that:

- Before setting up the post function, a custom text field is needed for storing the email addresses
- `%{nnnnn}` has to be replaced with the field code of this custom field containing the email addresses
- Like always, post functions within the **Create transition** have to be placed **after the Creates the issue originally** one

Once configured, the transition will look like this:

Create

TO DO

This is the **initial** transition in the workflow.

Screen: None - initial transition does not have a view.

Validators [Post Functions](#)

1

4

The following will be processed after the transition occurs

[Add post function](#)

1. Creates the issue originally.

2. **Update issue fields**

Target issues:

Current issue

Target fields and Source values:

Target Field	Type of Value	Source Value	Don't Overwrite
New watchers	Parsed text (advanced mode)	toString(textOnStringList(toStringList(% CF TEXT FIELD (SINGLE LINE), ", "), toString(usersWithEmail(^%))))	

This feature will be run as user in field **Current user:** [\(by user\)](#)

3. Re-index an issue to keep indexes in sync with the database.

4. Fire a **Issue Created** event that can be processed by the listeners.

Other examples of that function

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

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
- Setting the priority depending on the multiplication of custom fields
 - example
 - calculated-field
 - post-function
- Parse Email addresses 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