Create issue links based on a custom field value avoiding duplicates

On this page

Features used to implement the example

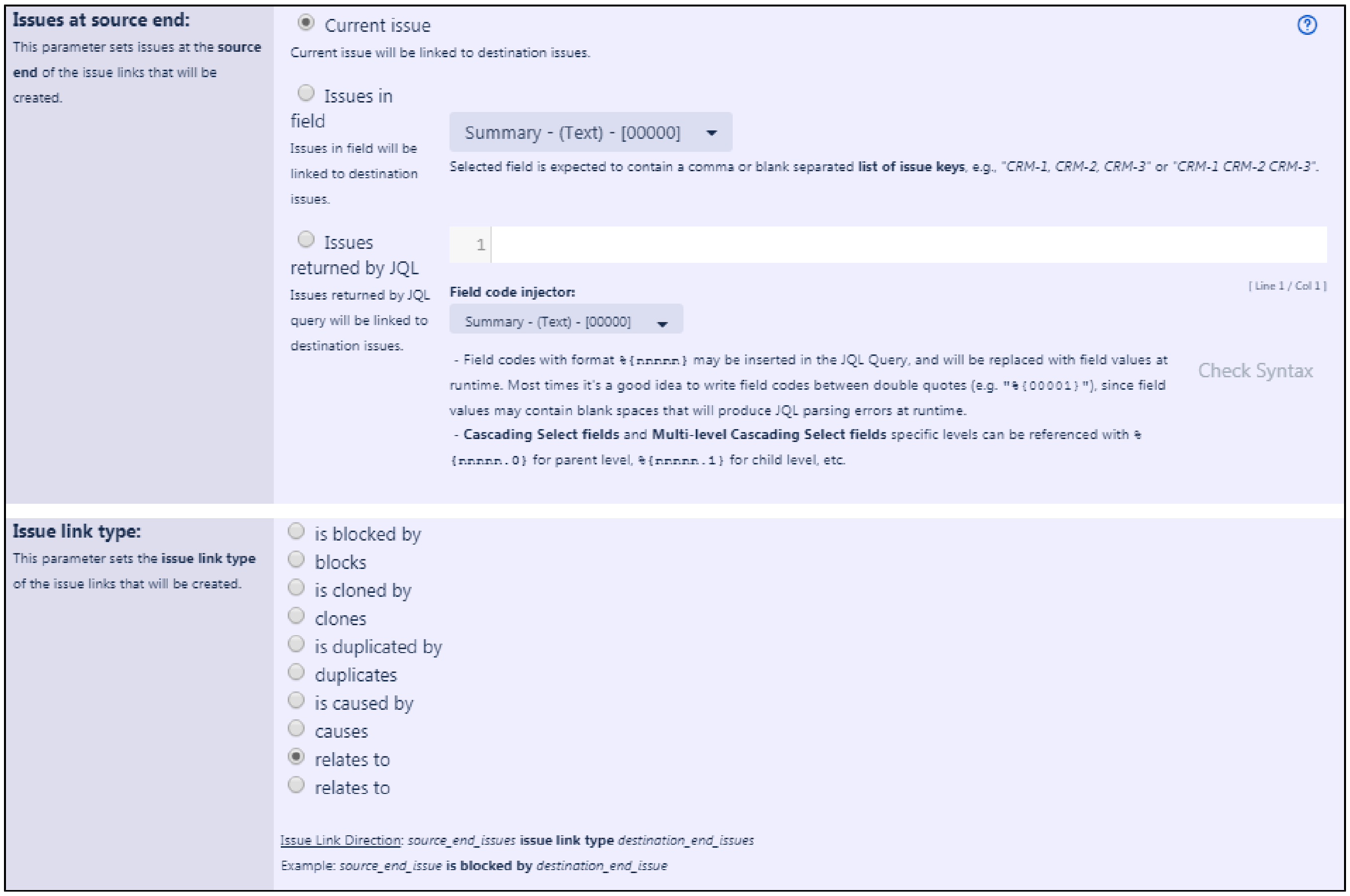
- **Create issue link**

Example: Create issue links based on a custom field value avoiding duplicates

In this example, issue links between issues with the same custom field value should be added avoiding duplicate links, e.g.

Given three issues with the same value for the custom field **"Custom-ID"**, Ticket A, Ticket B and Ticket C. When executing the transition on Ticket A, Ticket B and C should be linked to Ticket A. When now executing the transition on Ticket B, only Ticket C should be linked to Ticket B since Ticket A was already linked in the previous transition execution. In order to add issue links on a transition, the **Create issue link** post function is used.

The configuration will look like this:



Issues at source end:
This parameter sets issues at the **source end** of the issue links that will be created.

[✓ Current issue]
Current issue will be linked to destination issues.

Issues in field
Issues in field will be linked to destination issues.
Summary - (Text) - [00000]
Selected field is expected to contain a comma or blank separated **list of issue keys**, e.g., *"CRM-1, CRM-2, CRM-3"* or *"CRM-1 CRM-2 CRM-3"*.

Issues returned by JQL
Issues returned by JQL query will be linked to destination issues.

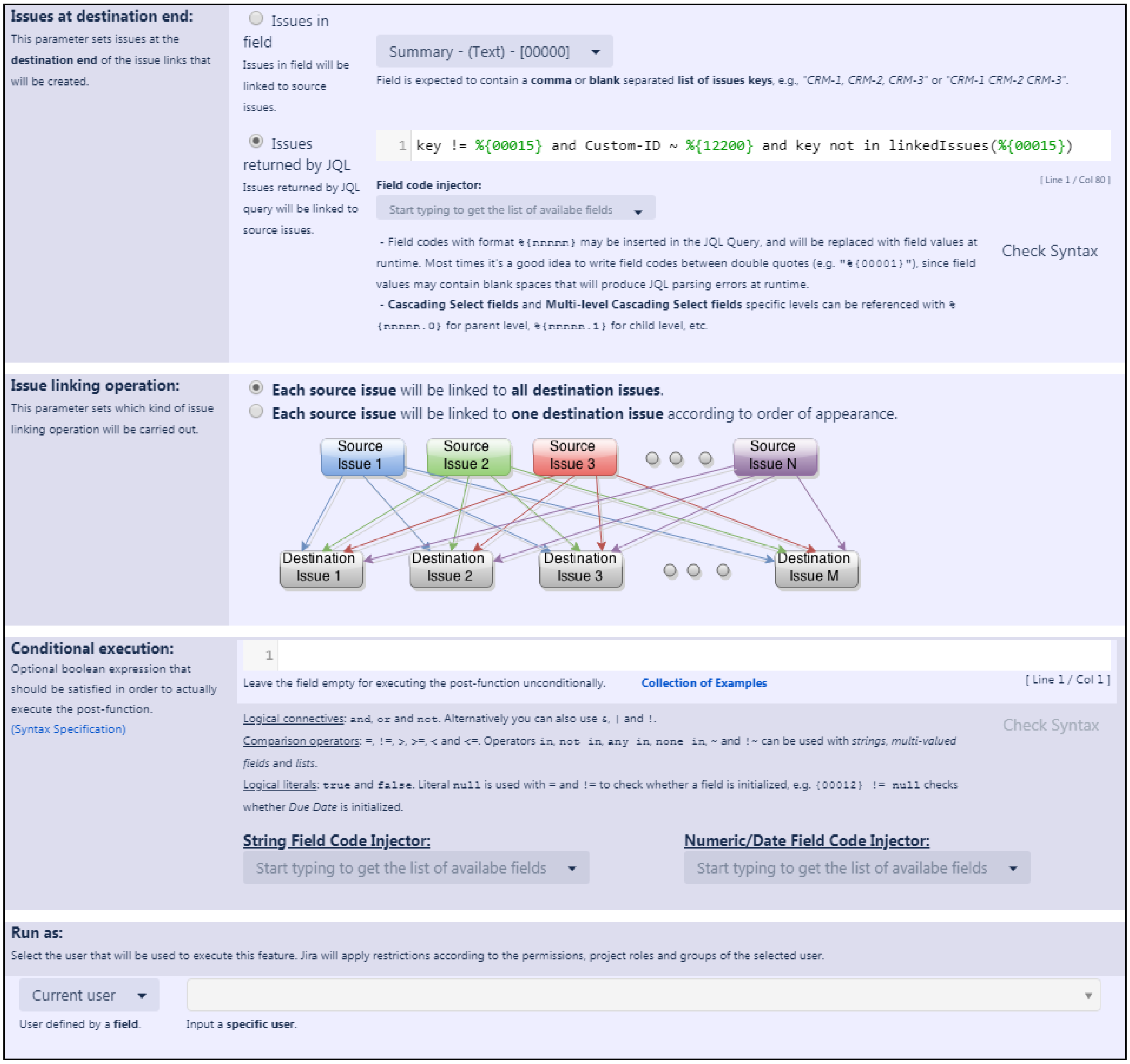
Field code injector:
Summary - (Text) - [00000]

- Field codes with format `%{nnnnn}` may be inserted in the JQL Query, and will be replaced with field values at runtime. Most times it's a good idea to write field codes between double quotes (e.g. `"%{00001}"`), since field values may contain blank spaces that will produce JQL parsing errors at runtime.
- **Cascading Select fields** and **Multi-level Cascading Select fields** specific levels can be referenced with `%{nnnnn.0}` for parent level, `%{nnnnn.1}` for child level, etc.

Issue link type:
This parameter sets the **issue link type** of the issue links that will be created.

is blocked by
blocks
is cloned by
clones
is duplicated by
duplicates
is caused by
causes
[✓ relates to]
relates to

Issue Link Direction: *source_end_issues* **issue link type** *destination_end_issues*
Example: *source_end_issue* **is blocked by** *destination_end_issue*



Issues at destination end:
This parameter sets issues at the **destination end** of the issue links that will be created.

( ) Issues in field
Issues in field will be linked to source issues.

Summary - (Text) - [00000]

Field is expected to contain a **comma** or **blank** separated **list of issues keys**, e.g., *"CRM-1, CRM-2, CRM-3"* or *"CRM-1 CRM-2 CRM-3"*.

[✓ Issues returned by JQL]
Issues returned by JQL query will be linked to source issues.

```
key != %{00015} and Custom-ID ~ %{12200} and key not in linkedIssues(%{00015})
```

[Line 1 / Col 80]

Field code injector:
Start typing to get the list of availabe fields

- Field codes with format `%{nnnnn}` may be inserted in the JQL Query, and will be replaced with field values at runtime. Most times it's a good idea to write field codes between double quotes (e.g. `"%{00001}"`), since field values may contain blank spaces that will produce JQL parsing errors at runtime.
- **Cascading Select fields** and **Multi-level Cascading Select fields** specific levels can be referenced with `%{nnnnn.0}` for parent level, `%{nnnnn.1}` for child level, etc.

Check Syntax

Issue linking operation:
This parameter sets which kind of issue linking operation will be carried out.

[✓ **Each source issue** will be linked to **all destination issues**.]
( ) **Each source issue** will be linked to **one destination issue** according to order of appearance.

Conditional execution:
Optional boolean expression that should be satisfied in order to actually execute the post-function.
(Syntax Specification)

Leave the field empty for executing the post-function unconditionally. **Collection of Examples** [Line 1 / Col 1]

Logical connectives: `and`, `or` and `not`. Alternatively you can also use `&`, `|` and `!`.
Comparison operators: =, `!=`, >, >=, < and <=. Operators `in`, `not in`, `any in`, `none in`, ~ and `!~` can be used with *strings, multi-valued fields* and *lists*.
Logical literals: `true` and `false`. Literal `null` is used with = and `!=` to check whether a field is initialized, e.g. `{00012} != null` checks whether *Due Date* is initialized.

Check Syntax

String Field Code Injector:
Start typing to get the list of availabe fields

Numeric/Date Field Code Injector:
Start typing to get the list of availabe fields

Run as:
Select the user that will be used to execute this feature. Jira will apply restrictions according to the permissions, project roles and groups of the selected user.

Current user
User defined by a **field**.

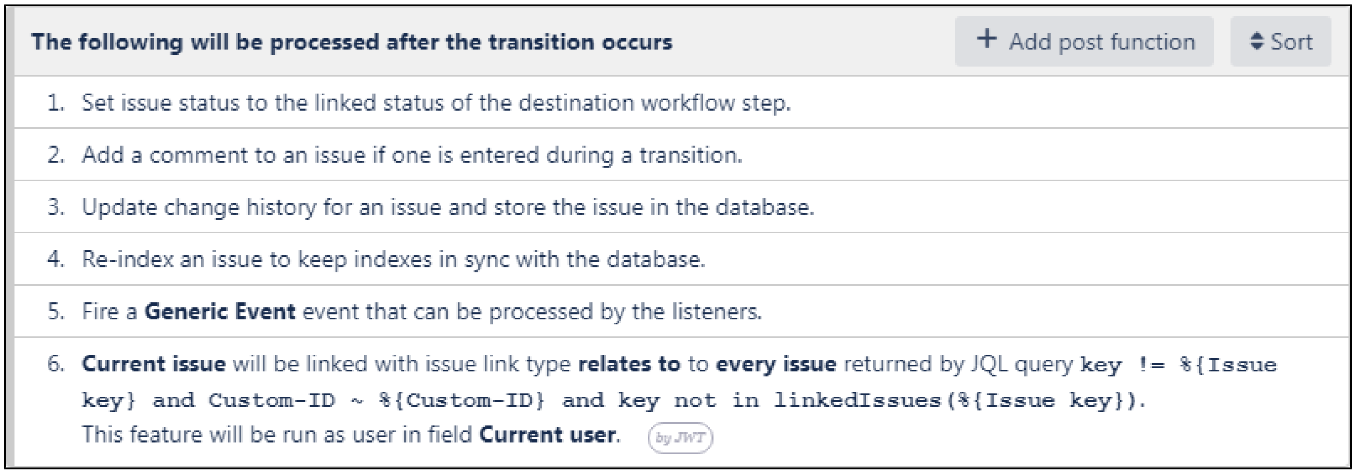
Input a **specific user**.

Issues returned by JQL expression is: **`key != %{00015} and Custom-ID ~ %{12200} and key not in linkedIssues(%{00015})`**

ℹ Note that:

- **%{00015}** is the field code for **"Issue key"**
- **%{12200}** is the field code for the custom field **"Custom-ID"** (this field code might differ on your instance)

Once configured, the transition will look like this:



1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. Re-index an issue to keep indexes in sync with the database.
5. Fire a **Generic Event** event that can be processed by the listeners.
6. **Current issue** will be linked with issue link type **relates to** to **every issue** returned by JQL query `key != %{Issue key} and Custom-ID ~ %{Custom-ID} and key not in linkedIssues(%{Issue key})`. This feature will be run as user in field **Current user**.

See **Result screenshots "Create issue links based on a custom field value avoiding duplicates"**

Other examples of that function

Page: Automatically create an issue link after issue creation on email by "Enterprise Mail Handler for Jira" app
Page: Create issue links based on a custom field value avoiding duplicates
Page: Creating issue links to issues with the same "Summary"
Page: Parse description for creating issue links
Page: Replace certain issue link types with different ones

Related Usage Examples

- Add and remove a single or a set of items from multi valued fields
  - example
  - post-function
  - custom-field
  - issue-links
  - sub-task
- Writing a comment to blocked issues when blocking issues are resolved
  - example
  - post-function
  - issue-links
- Transition linked issues in currently active sprint
  - example
  - post-function
  - issue-links
  - transition
- Automatically become watcher of every issue blocking an issue assigned to you
  - example
  - post-function
  - issue-links
- Make linked issues, sub-tasks and JQL selected issues progress through its workflows
  - example
  - condition
  - validator
  - post-function
  - issue-links
  - sub-task
  - transition
- Make an issue inherit highest priority among those of linked issues
  - example
  - post-function
  - issue-links
- Propagate highest priority from blocked issues to blocking issues
  - example
  - post-function
  - issue-links
- Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
  - example
  - post-function
  - custom-field
  - issue-links
- Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
  - example

    - post-function
    - issue-links
    - sub-task
    - work-log
- Parse description for creating issue links
    - example
    - post-function
    - issue-links
- Creating issue links to issues with the same "Summary"
    - example
    - post-function
    - issue-links
- Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC"
    - example
    - post-function
    - issue-links
    - sub-task
    - work-log
- Replace certain issue link types with different ones
    - example
    - post-function
    - issue-links
- Automatically create an issue link after issue creation on email by "Enterprise Mail Handler for Jira" app
    - example
    - post-function
    - issue-links
- Validate only issue links created in transition screen
    - example
    - validator
    - issue-links