

Creating issue links to issues with the same "Summary"

On this page

- [Features used to implement the example](#)
- [Example: Creating issue links to issues with the same "Summary"](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Create issue link](#)
-

Example: Creating issue links to issues with the same "Summary"

I want to link the newly created issues to the existing with the same "**Summary**" by transition. I've tried with your description for [Create issue link](#) for [Jira Workflow Toolbox](#) post-functions, but can't find the right settings. Hope you can give me advise.

You should use post-function [Create issue link](#) with the following configuration:

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]

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.

1

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
☒ **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

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.

1 summary ~ "%{00000}"

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 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.

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 certain **fixed user**. Expected value is a **user name**, don't confuse it with user's full name.

JQL Query used is:

summary ~ "%{00000}"

If you want to restrict issues within the same project use the following JQL query:

summary ~ "%{00000}" and project = "%{00018}"

Note that:

- %{00000} is field code for "Summary"
- %{00018} is field code for "Project key"

Once configured, transition "Create Issue" 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 1

Post Functions 4

The following will be processed after the transition occurs

Add post function

1. Creates the issue originally.

2. **Current issue** will be linked with issue link type **relates to** to **every issue** returned by JQL query **summary ~ "%{Summary}"**.
This feature will be run as user in field **Current 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: [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

- [Validate only issue links created in transition screen](#)
 - [example](#)
 - [validator](#)
 - [issue-links](#)
- [Require issue link when resolving as duplicate](#)
 - [example](#)
 - [validator](#)
 - [issue-links](#)
- [Ensure that all issues linked with a certain issue link type have "Due Date" field set](#)
 - [example](#)
 - [validator](#)
 - [issue-links](#)
- [Block an epic's transition depending on linked issues status and due date](#)
 - [example](#)
 - [validator](#)
 - [issue-links](#)
 - [transition](#)
- [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
- Prevent issue from moving forward if it's dependent on non-accepted tickets
 - example
 - validator
 - issue-links
 - transition
- Enforce linked issues in a specific project to be "Closed" before closing issue
 - example
 - validator
 - issue-links
 - transition
- Block or hide a transition for an issue depending on its issue links
 - example
 - validator
 - issue-links
 - transition
- Prevent transitioning when there is a blocking issue
 - example
 - validator
 - issue-links
 - sub-task
 - transition
- Prevent issue from being "Closed" if blocking issues aren't yet closed
 - example
 - validator
 - issue-links
 - transition
- Block creation of issue type X if it has not been linked with link type Y to issue type Z on the "Create Issue" screen
 - example
 - validator
 - issue-links
- Prevent issue from being closed if it has links of type "is blocked by" to open issues
 - example
 - condition
 - validator
 - issue-links
 - transition
- 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