

Replace certain issue link types with different ones

On this page

- [Features used to implement the example](#)
- [Example: Replace certain issue link types with different ones](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Create issue link](#)
- [Break issue link](#)
- [Copy parsed text to a field](#)

Example: Replace certain issue link types with different ones

The following situation:

- Two issue type: **Test** and **Story**
- Two link type: **Pruebas** (Es probado por, Esta probando a) and **Pruebas Old** (Fue probado por, Probó a)
- Pruebas Link type between Test and Story. "**Es probado por**" from Story and "**Esta probando a**" from Test.

I need change **Pruebas** link to **Pruebas Old**. For example:

Issue link before transition:

- Test A "**Esta probando a**" Story A
- Story A "**Es probado por**" Test A

Issue link after transition:

- Test A "**Probó a**" Story A
- Story A "**Fue probado por**" Test A

You should insert in the transition of Test's workflow the following 3 post-functions:

Post-function [Copy parsed text to a field](#) with the following configuration:

Target field:

Ephemeral string 1 - [Text]

Field to be written with the resulting parsed text.
☐ Don't overwrite target field if it's already set.

Parsing Mode:

☐ Basic
Basic mode: Insert field codes anywhere in the text, and they will be replaced with corresponding field values. Field code formats are `%(nnnnn)`, and `%(nnnnn.i)` for Cascading Select fields (i = 0 for base level).

☒ Advanced
Advanced mode: Strings literals are written in double quotes (*"This is a string."*). Operator '+' is used to concatenate strings, and field codes are like in basic mode, e.g., *"Issue key is " + %(00015) + "."*. More information at [parser syntax documentation](#).

Text to be parsed and then copied to target field: [Syntax Specification](#)

```
1 toString(linkedIssues("Esta probando a"))
```

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.	<div><input checked="" type="radio"/> Current issue Current issue will be linked to destination issues.</div> <div><input type="radio"/> Issues in field Issues in field will be linked to destination issues. 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".</div> <div><input type="radio"/> Issues returned by JQL Issues returned by JQL query will be linked to destination issues.</div>	<div>Summary - [Text]</div> <div>1</div> <div>Field code injector: Summary - [Text] - %{00000}</div> <div>- 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.</div> <div>- 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.</div>
Issue link type: This parameter sets the issue link type of the issue links that will be created.	<div><input type="radio"/> is blocked by</div> <div><input type="radio"/> blocks</div> <div><input type="radio"/> is cloned by</div> <div><input type="radio"/> clones</div> <div><input type="radio"/> is duplicated by</div> <div><input type="radio"/> duplicates</div> <div><input type="radio"/> Fue probado por</div> <div><input checked="" type="radio"/> Probó a</div> <div><input type="radio"/> Es probado por</div> <div><input type="radio"/> Esta probando a</div> <div><input type="radio"/> relates to</div> <div><input type="radio"/> relates to</div> <div><input type="radio"/> is validated by</div> <div><input type="radio"/> validates</div>	<div>Issue Link Direction: source_end_issues issue link type destination_end_issues</div> <div>Example: source_end_issue is blocked by destination_end_issue</div>
Issues at destination end: This parameter sets issues at the destination end of the issue links that will be created.	<div><input checked="" type="radio"/> Issues in field Issues in field will be linked to source issues.</div> <div><input type="radio"/> Issues returned by JQL Issues returned by JQL query will be linked to source issues.</div>	<div>Ephemeral string 1</div> <div>1</div> <div>Field code injector: Summary - [Text] - %{00000}</div> <div>- 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.</div> <div>- 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.</div>

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

-
- The diagram shows a bipartite graph structure. The top row contains 'Source Issue 1' (blue), 'Source Issue 2' (green), 'Source Issue 3' (red), and 'Source Issue N' (purple). The bottom row contains 'Destination Issue 1' (grey), 'Destination Issue 2' (grey), 'Destination Issue 3' (grey), and 'Destination Issue M' (grey). Directed edges connect source issues to destination issues: blue edges from Source 1, green from Source 2, red from Source 3, and purple from Source N. The edges are distributed across the destination issues, with some overlapping colors on a single edge.

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.

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

If all issue link types are kept unchecked, there won't be applied any filter by issue link type, i.e., all issue link types will be considered as selected.

Issue types for the linked issues:
Issue types required for issues at the other end of the issue link. Only if issue at the other end of the link belongs to one of selected issue types, the issue link will be broken.

☐ Access
☐ Bug
☐ Change
☐ Epic
☐ Fault
☐ IT Help
☐ Improvement
☐ New Feature
☐ Purchase
☒ Story
☐ Test
☐ Sub-task
☐ Technical task

If all issue types are kept unchecked, there won't be applied any filter by issue type, i.e., all issue types will be considered as selected.

Statuses for the linked issues:
Statuses required for issues at the other end of the issue link. Only if issue at the other end of the link is in one of selected statuses, the issue link will be broken.

☐ Open
☐ In Progress
☐ Reopened
☐ Resolved
☐ Closed
☐ Assigned
☐ Waiting for support
☐ Waiting for customer
☐ Waiting for Triage

If all statuses are kept unchecked, there won't be applied any filter by issue status, i.e., all statuses will be considered as selected.

Break only links within current project:

☐

When checked, issues at both sides of the link should belong to the same project, otherwise, no condition will be required in relation to project belonging.

Linked issues are in:
This is an **optional parameter** for setting which issues are at the **destination end** of the issue links which are going to be broken.

☐ Enable issue filtering at destination end
☐ Issues in field
☐ Issues returned by JQL

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".

Field code injector:

- 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.

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.

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.

Once configured, the transition in Test's workflow will look like this:

Triggers 0	Conditions 0	Validators 0	Post Functions 8
------------	--------------	--------------	------------------

The following will be processed after the transition occurs

Add post function

- The following text parsed in **advanced** mode will be copied to **Ephemeral string 1**:
`toString(linkedIssues("Esta probando a"))`
This feature will be run as user in field **Current user**.
- Current issue** will be linked with issue link type **Probó a** to **every issue** in field **Ephemeral string 1**.
This feature will be run as user in field **Current user**.
- Break issue links of **current issue** fulfilling the following conditions:
Outward issue link types: **Esta probando a**.
Issue types: **Story**.
Statuses: **any**
Linked issues can belong to **any** project.
This feature will be run as user in field **Current user**.

Other examples of that functions

Create issue link

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

Break issue link

Page: Replace certain issue link types with different ones

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
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

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

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](#)

- [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](#)
- [Make an issue inherit highest priority among those of linked issues](#)
 - [example](#)
 - [post-function](#)
 - [issue-links](#)