Linked Transition Post-Function

JSU for Jira Cloud

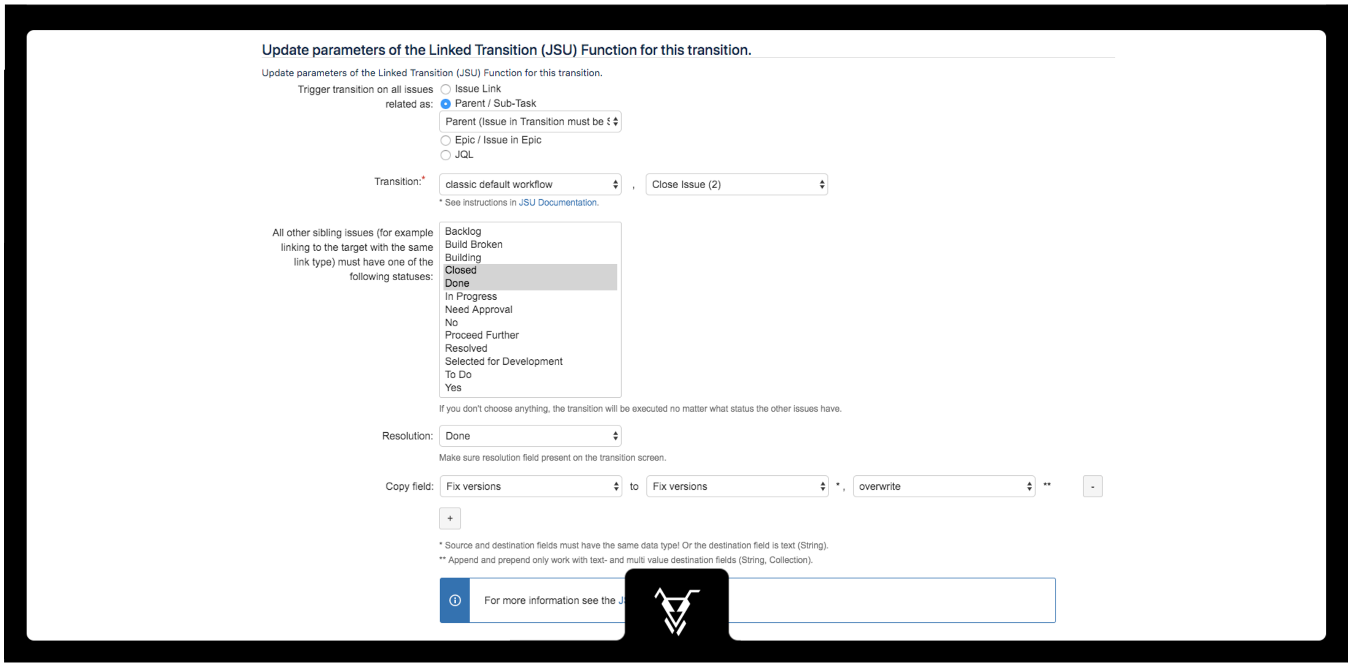
This is the documentation of JSU for Jira Cloud. If you are using JSU on Jira Server or on Jira Data Center, you can find the documentation here.

Description

This post function will trigger a transition on a related issue. It can be very powerful in conjunction with the Create a Linked Issue post function to 'connect' the workflows of 2 issues.

You can also specify any number of fields, that will be copied to the related issue.

Configuration



Update parameters of the Linked Transition (JSU) Function for this transition.

Update parameters of the Linked Transition (JSU) Function for this transition.

Trigger transition on all issues related as: ( ) Issue Link [✓ Parent / Sub-Task] ( ) Epic / Issue in Epic ( ) JQL

Parent (Issue in Transition must be S

Transition:* classic default workflow , Close Issue (2)

* See instructions in JSU Documentation.

All other sibling issues (for example linking to the target with the same link type) must have one of the following statuses:

Backlog
Build Broken
Building
[✓ Closed]
[✓ Done]
In Progress
Need Approval
No
Proceed Further
Resolved
Selected for Development
To Do
Yes

If you don't choose anything, the transition will be executed no matter what status the other issues have.

Resolution: Done

Make sure resolution field present on the transition screen.

Copy field: Fix versions to Fix versions * , overwrite **

* Source and destination fields must have the same data type! Or the destination field is text (String).

** Append and prepend only work with text- and multi value destination fields (String, Collection).

For more information see the J

Trigger transition on all issues related as

You have several different options, on which related issue the transition will be triggered.

Several of JSU's workflow modules provide the option the define the scope of the module on some related issues.

For example instead of copying a field within an issue during a post function, you might choose to copy it to a sub-task.

Types of Issue Relations

Related issue will be found by one of the following Jira concepts:

- **Issue Link**
  You can define the link type to define which issues exactly will be affected by the operation.
  If link type is ANY, the operation will be performed on any linked issues.
- **Parent / Sub-Task**
  The related issue is either parent or sub-task of each other.
- **Epic / Issue in Epic**
  This is only applicable, if you have Jira Software installed.
  The other issue is either the epic related by an epic link, or it is part of an epic.
- **JQL**
  A JQL query will be executed to find the issues, which will be affected by the post function. You can use some placeholders in the JQL query, which will be replaced with current field values of the issue in transition. For writing the JQL query please follow the instructions on JQL Reference

  Also see our JQL Use Cases for some examples.

Issue in Transition

We use the term 'Issue in Transition' when we refer to the issue for which a workflow condition is checked, for which a workflow validator is examined, or for which a workflow post function is performed.

Or in other words that issue, which triggered a workflow condition / validator / post function to be executed.

Source and Destination

For example the Copy Value From Other Field Post-Function allows you to define the issue in transition as the source or destination of the copy operations, while you define the other end with an issue relation.
The field value will then be read from the source issue and be written to the destination issue.

Other workflow modules do not have source and destination. So you just define the issue relation, which will apply for that workflow module.
For example Create a Linked Issue Post-Function will just create a new issue and then connect it with some Issue Relation to the issue in transition.

Restrictions

Please use some common sense what can and what cannot work.

Some examples:

- Copy Value From Other Field Post-Function there should be only 1 source issue. Otherwise it is not clear from which the value will be read. (The current implementation will just pick one of them and ignore the rest. Which one is not defined - sort of random.)
- If you use Create a Linked Issue Post-Function to create a new sub-task, you must also configure this post function to create the new issue as an issue type of a sub-task and make sure the target project is the same as the one of the issue in transition.

We try to perform whatever is possible. In case of problems, the workflow module will still run through, but write a message into the log file. See Troubleshooting how to turn on the full logging for JSU. It might be a good idea to do this while you test any new workflow configuration to make sure you're not missing any thing.

Transition

When the post function is performed, it will trigger the transition with that particular id (31 in the example above) on the linked issues.

When the post function is executed, it will just look if that transition id is available on the target issue. No matter what workflow and transition name you picked in the configuration screen (that is only to make it easier to find a particular transition id during configuration).

If the transition with that id is not available on the linked issue (probably because it is in a different status) nothing will happen. Also no comment or fields are copied, no resolution is set.

It is important to keep this in mind and design your workflows accordingly to prevent them to become sort of 'out of sync'. You might also use several 'Linked Transition' post functions in the same transition, each calling a different target transition, to match possible different statuses of the linked issue.

Also be aware, that there might be some workflow conditions or validators, which could prevent a transition to be performed.

The transition on the linked issue will be performed as the same user, who triggered the transition on the origin issue. If he does not has the necessary permissions, nothing will happen.

Resolution

It is important to following in mind:

1) To set the resolution, It is necessary to have resolve screen on transition. For more information, Please see https://confluence.atlassian.com/jirakb/mapping-a-screen-to-a-workflow-transition-720634253.html

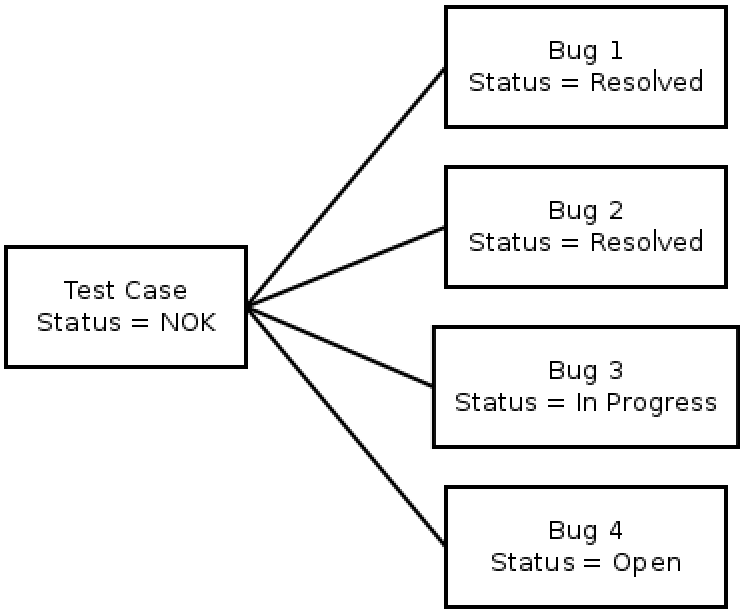
2) If any transition screen contains the resolution field, that field becomes mandatory. Here you have a chance to set a value for the resolution to be able to perform that transition.

Resolution here in the Post Function Configuration	**Resolution on the issues at the beginning when the transition is performed**	**What happens?**
empty	(does not matter)	The resolution on the issue won't be changed.
Any value	Any other value	The resolution will be set to the defined value.

All other sibling issues must have one of the following statuses

This allows you, that only 'the last' issue will trigger a linked transition.

Consider the following structure of linked issues from our Linked Transition Post-Function use case:



A test case issue has several linked bugs. The bugs are linked as 'from test' to the test case.

Only when the the last bug is fixed, the test case should be set to the Status 'Ready for Re-Test' (this is the transition 'All bugs fixed (51)' in the above screenshot).
The prevent this transition to be executed already when the first bug is fixed, we configure here two statuses

```
All other issues linking to the target must have one of the following statuses: Resolved, Closed
```

(Resolved is not visible in the above screenshot, but actually selected further down in the list of statuses.)

If bug 3 in the above diagram is resolved, nothing will happen, because bug 4 still has another status (than Resolved or Closed). Finally after that, bug 4 is resolved - all bugs now have status Resolved (Closed would also be ok) - now the transition ('All bugs fixed (51)') on the test case is executed to set it to 'Ready for Re-Test'.

Copy Fields

This fields will be copied to the linked issue, after the transition has been performed.
The conditions and validators of your linked transition will still use the old field values.

You can copy the value form some source field to some destination field. Click the + Button to add additional field pairs to your configuration.

Keep in mind that not all conversions from source to destination field are supported, nor feasible. We can only ensure, that it will work if source and destination fields are of the same field type, or the destination field is a text field. (Still, some additional combinations of different field types might work as well - although not 'officially' supported.)

Overwrite / Append / Prepend

For text fields and some field that can take multiple values (like checkboxes) you can choose to overwrite, append or prepend the new value to any existing value. In the case of text field, you can also choose a separator that will be put between the values (not shown in the screen shot above).

Special 'Sources'

As source you have some additional options:

- *** empty ***
  The destination field will be cleared.

Supported Field Types

In its different modules (especially those for workflows), the JSU app supports many different field types. System fields, as well as custom fields.

However you should be aware, that not all field types are supported. Also not in all combinations. We think we cover the most important field types and still are continuously adding and improving which and how different field types are supported. But the one you need, might just not (yet) work. Some custom fields of other third party app might never get supported.

For that reason you should always test anything you do with the JSU app with fields. Before you buy a license for JSU, try it with a free evaluation license, if it works for you.

Asynchronous Execution

There is a fundamental differences between Jira Server/Data Center and Jira Cloud: In Jira Cloud **post functions of apps are executed asynchronous** as a **'background job'**.

- This means a post function will be executed after the transition has completed on Jira Cloud.
- There is no guaranteed order in which several post functions are performed. They might be executed in a different order than the one you had set them up in Jira's workflow configuration. So you cannot rely on any result of a 'previous' post function. The order how the post functions are executed might be different every time the transition is performed.
- When a post function has a problem (due to misconfiguration or the current data of a Jira issue), the transition (and all other post functions) will be performed nevertheless.
- The result of a post function might not be visible immediately in the browser.
- There are no error messages displayed to the user.

Example

See the Use Case Testing and Fixing Bugs for a nice example, how several of our customers use it.

Troubleshooting

For general troubleshooting about JSU see Linked Transition .

If a linked transition does not get triggered, or it even block your origin transition, these are a few things to check:

- Did you check the log files. There are cases, when the linked transition is not performed silently. But you will find a message in the log files on the server. You might increase to logging level (as described in Linked Transition).
- What happens when you manually click the linked transition. Does it work, or might there be a problem?
- Does this problem only happen by another user? Check that the user performing origin transition, also has enough permissions for the linked transition.
- Is there a transition screen for the linked transition? Is there any condition, validator or post function on the linked transition? Could these prevent the transition to be performed in an automatised way (with the 'Linked Transition' post function)?
- In the order of all post functions of the orign transition, the 'Linked Transition' post function must be the last one.
- Did you try to set a particular Resolution in the linked transition. Or with 'Resolution=empty' - does it then work?
- Did you check already the documentation above? You might find antoher hint what to look for.