Propagate Any Fields

Usage cases

You want to propagate to linked issue, some of System or Custom Fields of the current Issue.

How it works

By defining this Workflow Function in a Transition of your workflow, you are able to force some system and custom fields of **Linked Issues** to be updated with supported field's value of current Issue.

Configuring the Feature

To configure the **Propagate System and Custom Fields to Linked Issue** Post-Function, please follow these steps :

- Since this Workflow Function does not need to be associated with a screen containing the concerned CustomFields, the function can be added in any Workflow Transition.
  It might be:
    - A Common Transition,
    - Or a Global Transition.

> ⊘ Note that if you need to use such Post-Function in a Initial Transition (Create Transition), you have to check that the Post-Function is used :
>
> - After **Create Issue** Post-Function
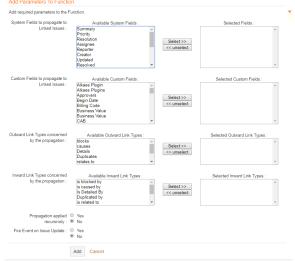> - and before **index Issue** Post-Function
>
> If it is not respected you will experience an error message when trying to perform the transition.

- Add the **Propagate System and Custom Fields to Linked Issue** Post-Function to any wanted transition.

  For more information on adding a Post Function in Workflow, please see JIRA: Adding a Post Function

○ System and Custom Fields Propagation to Sub Task ※	Propagate System and Custom Fields Values to Sub Task
○ System and Custom Fields Propagation to linked Issues ※	Propagate System and Custom Fields Values to linked Issues

- Its configuration allows you to specify which System and/or Custom Fields have to be inherited from the current Issue to linked Issues (Inward or Outward Links), recursively or not.

  You have to select
    - The list of **System Fields** concerned by the propagation,
    - The list of **Custom Fields** concerned by the propagation,
    - The list of **Outward Links** to follow for this propagation,
    - The list of **Inward Links** to follow for this propagation,

⊘

- If the propagation has to be applied **recursively**.

> ⊘ In case of Recursive Propagation, the **Propagate System and Custom Fields to Linked Issue** Workflow Function will take care that :
>   - The propagation is not done more than one time per Issue. The Recursivity skips to the next issue,
>   - The Recursivity is stopped as soon as an Issue has been nodified by a same Link Type (Case of circuit!)

Parameters	Description
Available System Fields	List of all System Field candidate for a propagation. Some of System Fields are filtered (See Unsupported Fields)
Selected System Fields	List of all selected System Fields for a propagation.
Available Custom Fields	List of all Custom Field candidate for a propagation. Some of Custom Fields are filtered, like all Calculated Custom Fields.
Selected Custom Fields	List of all selected Custom Fields for a propagation.
Available Outward Link Types	List of all Outward Link Types candidate for a propagation.
Selected Outward Link Types	List of all selected Outward Link Types for a propagation.
Available Inward Link Types	List of all Inward Link Types candidate for a propagation.
Selected Inward Link Types	List of all selected Inward Link Types for a propagation.
Propagation applied recursively	Has the propagation to be done recursively.

- You will end up with a transition looking like:
    1. 🎯 Propagate System and Custom Fields Values to linked Issues
    - Propagation applied recursively : **No**
    - Fire Event on Issue Update : **No**
    - Selected Custom Fields : **Business Value**,
    - Selected System Fields : **Reporter**,
    - Selected Outward Link Types : **relates to**,
    - Selected Inward Link Types : **Duplicated by**,

And the XML Declaration as follow :

```
<post-function >
        <function type="class">
          <arg name="customFieldsSelected">customfield_10070,</arg>
          <arg name="full.module.key">fr.alkaes.minyaa.jira-plugin-minyaa-
workflowspropagatefields-to-linked-issues-function</arg>
          <arg name="inwardLinkTypesSelected">10020,</arg>
          <arg name="fieldsSelected">reporter,</arg>
          <arg name="linkTypesSelected">10030,</arg>
          <arg name="class.name">fr.alkaes.myaawf.workflow.postfunction.
PropagateFieldToLinkedIssues</arg>
          <arg name="recursive">false</arg>
        </function>
</post-function>
```

Unsupported Fields

This Workflow Post-Function will assumes as **Unsupported Fields** for selection the following list of field :

- the Key fields:

- the Project Key (**com.atlassian.jira.issue.fields.ProjectSystemField**)
- the Issue Key (**com.atlassian.jira.issue.fields.ProjectSystemField**)
- The field managed by Schemes:
  - the Issue Type (**com.atlassian.jira.issue.fields.KeySystemField**)
  - the Issue Status (**com.atlassian.jira.issue.fields.StatusSystemField**)
- the calculated fields:
  - Fields marked as unavailable by FieldManager (TimeTracking, Agregate Time Tracking)
  - Work Ratio field
  - any Calculated Custom Field (**com.atlassian.jira.issue.customfields.impl. CalculatedCFType**)
  - any Minyaa Abstract Custom Fields (**com.atlassian.jira.issue.customfields.impl. AbstractLinkedFieldType**)
- Some other fields for which has no sense to propagate:
  - Created Date (**com.atlassian.jira.issue.fields.CreatedSystemField**)
  - Attachments (**com.atlassian.jira.issue.fields.ThumbnailSystemField**)
  - Thumbnails (**com.atlassian.jira.issue.fields.AttachmentSystemField**)
  - Sub Task Link (**com.atlassian.jira.issue.fields.SubTaskSystemField**)
  - Issue Link (**com.atlassian.jira.issue.fields.IssueLinksSystemField**)
  - Vote Counter (**com.atlassian.jira.issue.fields.VotesSystemField**)
  - Comments (**com.atlassian.jira.issue.fields.CommentSystemField**)

(i) **See also ...**

- Issue Validation
- Adding a Post Function
- Defining a Screen
- Configuring Workflow - Using a Screen
- Adding Custom fields