Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"

On this page

- Features used to implement the example
- Example: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
- Other examples of that functions
- Related Usage Examples

Features used to implement the example

- **Read fields from linked issues or sub-tasks**
- **Write field on linked issues or sub-tasks**
- **Set a field as a function of other fields**

Example: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"

I have the following use case: **Project A** & **Project B** are linked by issues to a **Project C. Project C** is linked by issues to **Project A**. First setting a date on B a date field of A must be set. Then when setting a date on C it must update the date on A **if its date is newer**. I thought first of copying within a transaction of B the date from A to an Ephemeral field and compare then the both dates. But there is no such field like 'Ephemeral Date'.

"**Ephemeral number fields**" can be used to store Date and Date-Time fields, since this kind of fields contain the number of milliseconds elapsed from 1/1/1970.

You can implement your requirements as I explain:

- Use post-function **Read fields from linked issues or sub-tasks** to get the newest date between issues in project B (or C) and in project A. You will need to check "**Read also current issue:**" and "**highest date**". Store the value in an "**Ephemeral number**" field.
- Use post-function **Write field on linked issues or sub-tasks** to write the value stored in the "**Ephemeral number**" field into date field in issue in project A.

The checkbox "**Read also current issue:**" does not work, because the field in **Project A** is named different as the fields in **Project B** and **C**. Is there a solution to compare two dates and setting the desired one?

Yes, you can do it, I'm going to explain what you have to implement in the workflow used by **project B** (for **project C** is exactly the same solution except for the name of the field). You don't have to do anything in the workflow used by project

Let's suppose that:

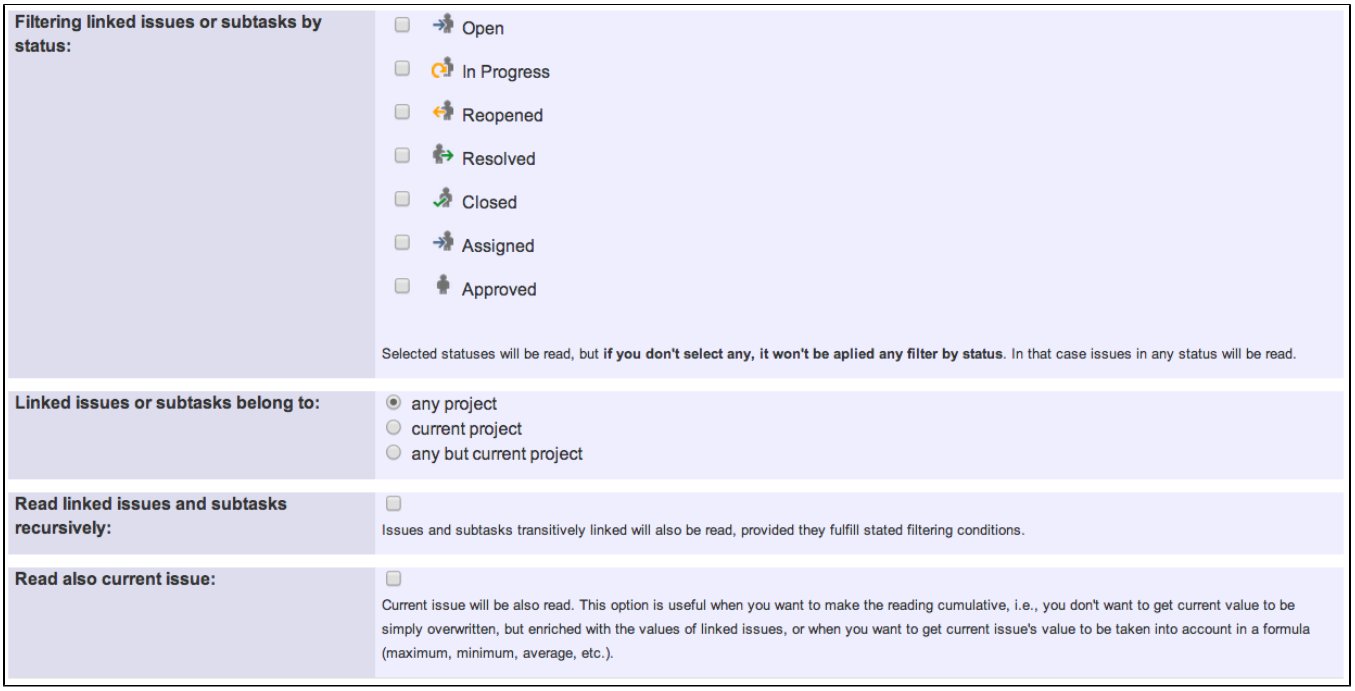
- Issue in **project B** is linked to issue in **project A** by a link of type "**blocks**", i.e., issue in project B **blocks** issue in project A
- Field in **project A** is called "**My Date**" and field in project B is "**Due date**".

The solution is as follows:

Use post-function **Read fields from linked issues or sub-tasks** to read the date field "**My Date**" in **project A** and store it in "**Ephemeral number 1**" in **project B**:

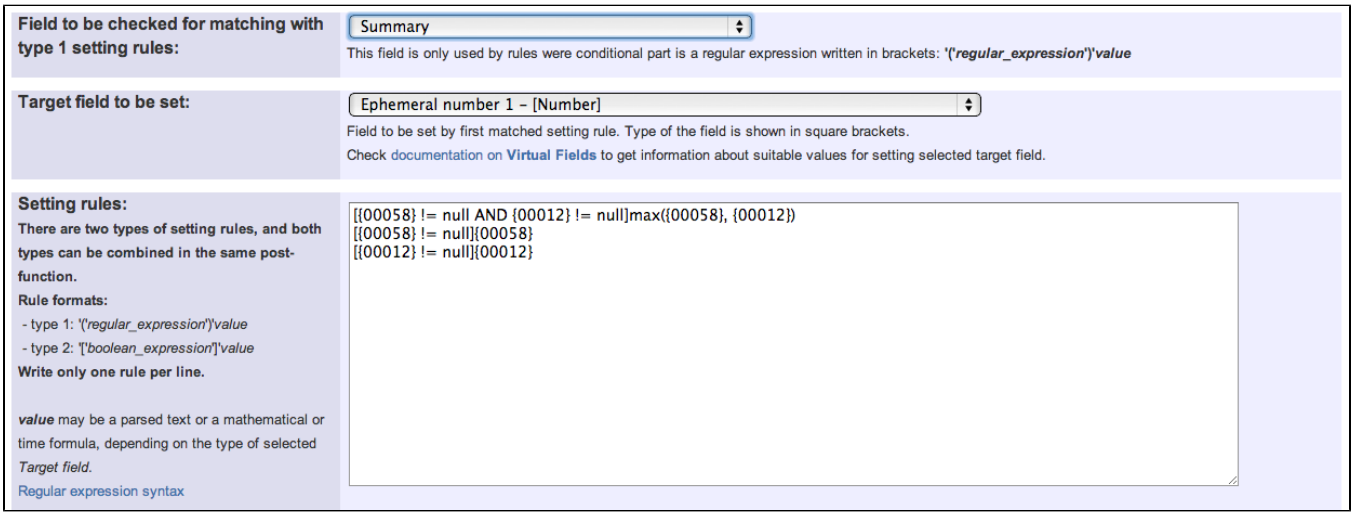
Source field to be read from linked issues or subtasks:	My Date
Target field to be written in current issue:	Ephemeral number 1
Value to write in target field in case source field is of type Number, Priority, Date, Date and Time:	lowest date
Filtering by inward issue link type:	☐ is blocked by ☐ is cloned by ☐ is duplicated by ☐ has Epic ☐ relates to Only issues linked to current issue by selected inward issue link types will be read.
Filtering by outward issue link type:	☑ blocks ☐ clones ☐ duplicates ☐ is Epic of ☐ relates to Only issues linked to current issue by selected outward issue link types will be read.

Read also subtasks fulfilling condition on issue type, status and project:	☐ This option only makes sense when current issue itself is not a subtask.
Read also sister subtasks fulfilling condition on issue type, status and project:	☐ Sister subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.
Filtering linked issues or subtasks by issue type:	☐ Bug ☐ Epic ☐ Improvement ☐ New Feature ☐ Order ☐ Story ☐ Task ☐ Agile Task ☐ Engineering Subtask ☐ QA Subtask ☐ Sub-task ☐ Testing subtask ☐ Validation subtask Selected issue types will be read, but **if you don't select any, it won't be aplied any filter by issue type**. In that case all the issue types will be read.



I'm supposing that you are reading only from one issue in **project A**, so it has no effect the option you select in parameter "Value to write in target field...". In the example I selected "**lowest date**".

Use post-function **Set a field as a function of other fields** to set "**Ephemeral number 1**" with newer date between "**Ephemeral number 1**" and "**Due date**". You should consider the cases where only "**Ephemeral number 1**" or only "**Due date**" are set:



Note that:

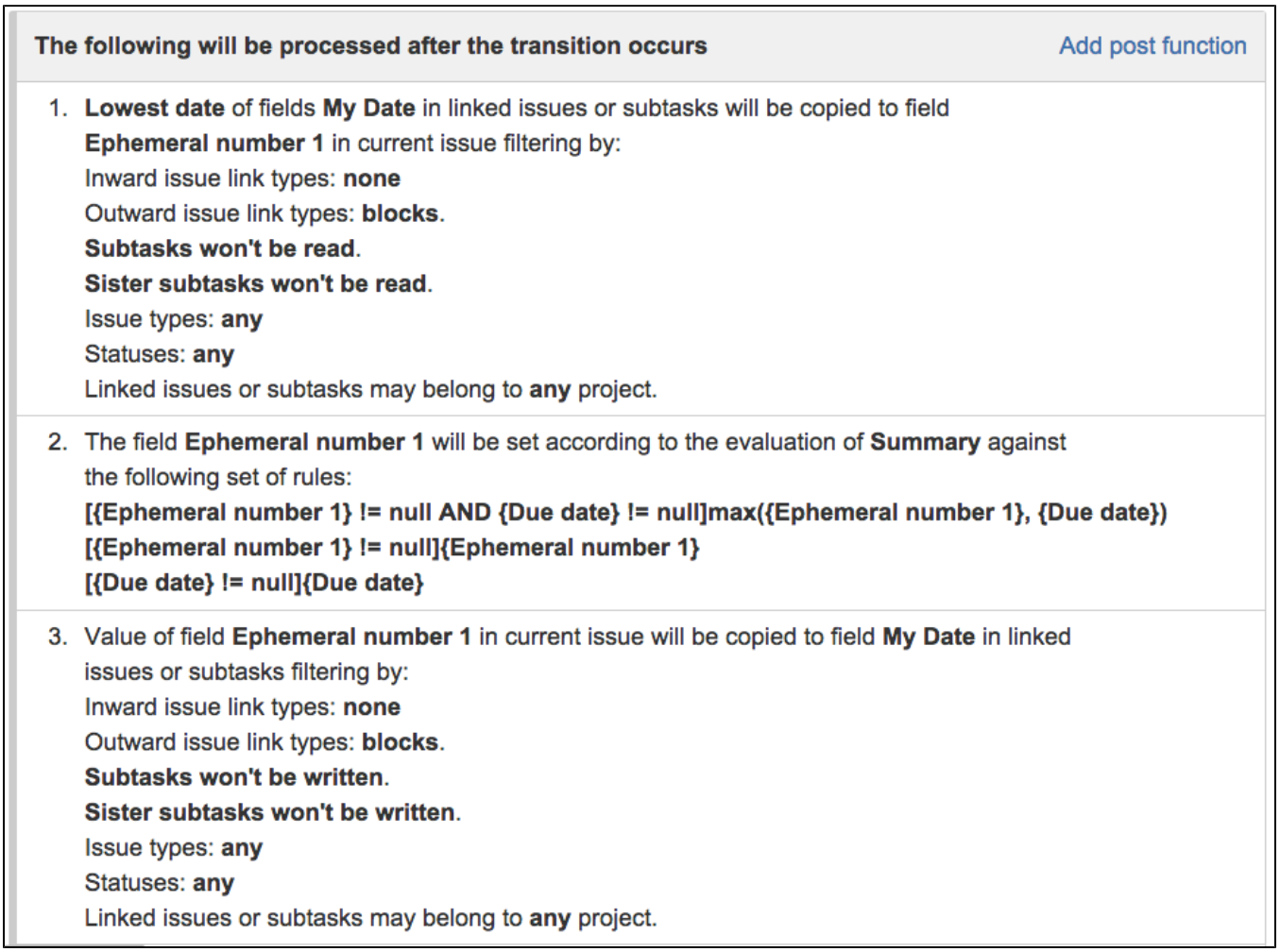
- **{00058}** is field code for **Ephemeral number 1**
- **{00012}** is field code for **Due date**

Use post-function **Write field on linked issues or sub-tasks** to write "**Ephemeral number 1**" in **project B** into field "**My Date**" in **project A**:

Source field to be read in current issue:	Ephemeral number 1
Target field to be written in linked issue or subtask:	My Date
Filtering by inward issue link type:	☐ is blocked by ☐ is cloned by ☐ is duplicated by ☐ has Epic ☐ relates to Only issues linked to current issue by selected inward issue link types will be written.
Filtering by outward issue link type:	☑ blocks ☐ clones ☐ duplicates ☐ is Epic of ☐ relates to Only issues linked to current issue by selected outward issue link types will be written.
Write also subtasks fulfilling condition on issue type, status and project:	☐ This option only makes sense when current issue itself is not a subtask.
Write also sister subtasks fulfilling condition on issue type, status and project:	☐ Sister subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.
Filtering linked issues or subtasks by issue type:	☐ Bug ☐ Epic ☐ Improvement ☐ New Feature ☐ Order ☐ Story ☐ Task ☐ Agile Task ☐ Engineering Subtask ☐ QA Subtask ☐ Sub-task ☐ Testing subtask ☐ Validation subtask Selected issue types will be written, but **if you don't select any, it won't be aplied any filter by issue type**. In that case all the issue types will be written.



Once configured, transition configuration will look like this:



Other examples of that functions

Read fields from linked issues or sub-tasks

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: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
Page: Copy attachments from one issue to another
Page: Make an issue inherit highest priority among those of linked issues
Page: Propagate highest priority from blocked issues to blocking issues
Page: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue

---

Write field on linked issues or sub-tasks

Page: Add and remove a single or a set of items from multi valued fields
Page: Automatically become watcher of every issue blocking an issue assigned to you
Page: Automatically close resolved sub-tasks when parent issue is closed
Page: Automatically resolve an epic when all its stories are resolved
Page: Compose dynamic text by inserting field values in a text template
Page: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
Page: Copy attachments from one issue to another
Page: Create a comment in sub-tasks when parent transitions
Page: Creating a Jira Service Desk internal comment
Page: Creating a Jira Service Desk internal comment on linked issues
Page: Execute transition in epic
Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows
Page: Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
Page: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
Page: Transition sub-tasks when parent is transitioned

Set a field as a function of other fields

Page: Add watcher depending on security level
Page: Add watchers based on issue type
Page: Add watchers depending on the value of a custom field
Page: Assign issue based on the value of a Cascading Select custom field
Page: Assign issue to a specific user based on a specific custom field value
Page: Assign issue to current user if assignee is empty
Page: Assign issue to current user if the user is not member of a certain project role
Page: Change assignee based on a custom field
Page: Change parent's status depending on sub-task's summary
Page: Changing issue priority depending on issue description
Page: Compose dynamic text by inserting field values in a text template
Page: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
Page: Limit the number of hours a user can log per day
Page: Make parent issue progress through its workflow
Page: Rise priority if due date is less than 3 weeks away
Page: Set "Due date" depending on the value of other fields, in case it's uninitialized
Page: Set "Due date" to a specific day of next week no matter of date of creation this week
Page: Set "Due date" to current date at issue creation if not initialized
Page: Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field
Page: Set a date based on current date
Page: Set a field based on reporter's email
Page: Set a watcher at ticket creation depending on custom field's value
Page: Set assignee depending on issue type
Page: Set security level based on groups and project roles the reporter or creator are in
Page: Set security level depending on reporter or creator
Page: Set the assignee based on a condition
Page: Set the value of a field of type "User Picker" depending on other field's value

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
- 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
- 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
- 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

Page: Set watchers depending on the value of a custom field
Page: Setting a custom field (User Picker) based on the value of another custom field (Text Field)
Page: Setting a field's default value depending on another field
Page: Setting the priority depending on the multiplication of custom fields
Page: Transition an issue automatically depending on the value of a field
Page: Unassign an issue when assigned to project leader
Page: Update checkboxes custom field if a file has been attached during a transition
Page: Using project properties to calculate custom sequence numbers

- issue-links
- transition

- Transition linked issues in currently active sprint
  - example
  - post-function
  - issue-links
  - transition