

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:	<div>My Date</div>
Target field to be written in current issue:	<div>Ephemeral number 1</div>
Value to write in target field in case source field is of type Number, Priority, Date, Date and Time:	<div>lowest date</div>
Filtering by inward issue link type:	<div><div><div><input type="checkbox"/> is blocked by</div><div><input type="checkbox"/> is cloned by</div><div><input type="checkbox"/> is duplicated by</div><div><input type="checkbox"/> has Epic</div><div><input type="checkbox"/> relates to</div></div><div>Only issues linked to current issue by selected inward issue link types will be read.</div></div>
Filtering by outward issue link type:	<div><div><div><input checked="" type="checkbox"/> blocks</div><div><input type="checkbox"/> clones</div><div><input type="checkbox"/> duplicates</div><div><input type="checkbox"/> is Epic of</div><div><input type="checkbox"/> relates to</div></div><div>Only issues linked to current issue by selected outward issue link types will be read.</div></div>
Read also subtasks fulfilling condition on issue type, status and project:	<div><div><input type="checkbox"/></div><div>This option only makes sense when current issue itself is not a subtask.</div></div>
Read also sister subtasks fulfilling condition on issue type, status and project:	<div><div><input type="checkbox"/></div><div>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.</div></div>
Filtering linked issues or subtasks by issue type:	<div><div><div><input type="checkbox"/>  Bug</div><div><input type="checkbox"/>  Epic</div><div><input type="checkbox"/>  Improvement</div><div><input type="checkbox"/>  New Feature</div><div><input type="checkbox"/>  Order</div><div><input type="checkbox"/>  Story</div><div><input type="checkbox"/>  Task</div><div><input type="checkbox"/>  Agile Task</div><div><input type="checkbox"/>  Engineering Subtask</div><div><input type="checkbox"/>  QA Subtask</div><div><input type="checkbox"/>  Sub-task</div><div><input type="checkbox"/>  Testing subtask</div><div><input type="checkbox"/>  Validation subtask</div></div><div>Selected issue types will be read, but if you don't select any, it won't be applied any filter by issue type. In that case all the issue types will be read.</div></div>

Filtering linked issues or subtasks by status:	<input type="checkbox"/> Open <input type="checkbox"/> In Progress <input type="checkbox"/> Reopened <input type="checkbox"/> Resolved <input type="checkbox"/> Closed <input type="checkbox"/> Assigned <input type="checkbox"/> Approved
<small>Selected statuses will be read, but if you don't select any, it won't be applied any filter by status. In that case issues in any status will be read.</small>	
Linked issues or subtasks belong to:	<input checked="" type="radio"/> any project <input type="radio"/> current project <input type="radio"/> any but current project
Read linked issues and subtasks recursively:	<input type="checkbox"/> <small>Issues and subtasks transitively linked will also be read, provided they fulfill stated filtering conditions.</small>
Read also current issue:	<input type="checkbox"/> <small>Current issue will be also read. This option is useful when you want to make the reading cumulative, i.e., you don't want to get current value to be simply overwritten, but enriched with the values of linked issues, or when you want to get current issue's value to be taken into account in a formula (maximum, minimum, average, etc.).</small>

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:














Field to be checked for matching with type 1 setting rules:	<div>Summary</div> <small>This field is only used by rules where conditional part is a regular expression written in brackets: <code>'(regular_expression)'value</code></small>
Target field to be set:	<div>Ephemeral number 1 - [Number]</div> <small>Field to be set by first matched setting rule. Type of the field is shown in square brackets. Check documentation on Virtual Fields to get information about suitable values for setting selected target field.</small>
Setting rules: <small>There are two types of setting rules, and both types can be combined in the same post-function. Rule formats: - type 1: <code>'(regular_expression)'value</code> - type 2: <code>'(boolean_expression)'value</code> Write only one rule per line. value may be a parsed text or a mathematical or time formula, depending on the type of selected Target field. Regular expression syntax </small>	<pre>{00058} != null AND {00012} != null max({00058}, {00012}) {{00058} != null {00058} {{00012} != null {00012}</pre>

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:	<div><div><input type="checkbox"/> is blocked by</div><div><input type="checkbox"/> is cloned by</div><div><input type="checkbox"/> is duplicated by</div><div><input type="checkbox"/> has Epic</div><div><input type="checkbox"/> relates to</div></div> <div>Only issues linked to current issue by selected inward issue link types will be written.</div>
Filtering by outward issue link type:	<div><div><input checked="" type="checkbox"/> blocks</div><div><input type="checkbox"/> clones</div><div><input type="checkbox"/> duplicates</div><div><input type="checkbox"/> is Epic of</div><div><input type="checkbox"/> relates to</div></div> <div>Only issues linked to current issue by selected outward issue link types will be written.</div>
Write also subtasks fulfilling condition on issue type, status and project:	<div><input type="checkbox"/></div> <div>This option only makes sense when current issue itself is not a subtask.</div>

Write also sister subtasks fulfilling condition on issue type, status and project:	<div><input type="checkbox"/></div> <div>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.</div>
Filtering linked issues or subtasks by issue type:	<div><div><input type="checkbox"/>  Bug</div><div><input type="checkbox"/>  Epic</div><div><input type="checkbox"/>  Improvement</div><div><input type="checkbox"/>  New Feature</div><div><input type="checkbox"/>  Order</div><div><input type="checkbox"/>  Story</div><div><input type="checkbox"/>  Task</div><div><input type="checkbox"/>  Agile Task</div><div><input type="checkbox"/>  Engineering Subtask</div><div><input type="checkbox"/>  QA Subtask</div><div><input type="checkbox"/>  Sub-task</div><div><input type="checkbox"/>  Testing subtask</div><div><input type="checkbox"/>  Validation subtask</div></div> <div>Selected issue types will be written, but if you don't select any, it won't be applied any filter by issue type. In that case all the issue types will be written.</div>

Filtering linked issues or subtasks by status:	<input type="checkbox"/> Open <input type="checkbox"/> In Progress <input type="checkbox"/> Reopened <input type="checkbox"/> Resolved <input type="checkbox"/> Closed <input type="checkbox"/> Assigned <input type="checkbox"/> Approved
<p>Selected statuses will be written, but if you don't select any, it won't be applied any filter by status. In that case issues in any status will be written.</p>	
Linked issues or subtasks belong to:	<input checked="" type="radio"/> any project <input type="radio"/> current project <input type="radio"/> any but current project
Write linked issues and subtasks recursively:	<input type="checkbox"/> <p>Issues and subtasks transitively linked will also be written, provided they fulfill stated filtering conditions.</p>

Once configured, transition configuration will look like this:

The following will be processed after the transition occurs	Add post function
<p>1. Lowest date of fields My Date in linked issues or subtasks will be copied to field Ephemeral number 1 in current issue filtering by:</p> <p>Inward issue link types: none</p> <p>Outward issue link types: blocks.</p> <p>Subtasks won't be read.</p> <p>Sister subtasks won't be read.</p> <p>Issue types: any</p> <p>Statuses: any</p> <p>Linked issues or subtasks may belong to any project.</p>	
<p>2. The field Ephemeral number 1 will be set according to the evaluation of Summary against the following set of rules:</p> <p>[[Ephemeral number 1] != null AND {Due date} != null]max({Ephemeral number 1}, {Due date})</p> <p>[[Ephemeral number 1] != null]{Ephemeral number 1}</p> <p>[[Due date] != null]{Due date}</p>	
<p>3. Value of field Ephemeral number 1 in current issue will be copied to field My Date in linked issues or subtasks filtering by:</p> <p>Inward issue link types: none</p> <p>Outward issue link types: blocks.</p> <p>Subtasks won't be written.</p> <p>Sister subtasks won't be written.</p> <p>Issue types: any</p> <p>Statuses: any</p> <p>Linked issues or subtasks may belong to any project.</p>	

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](#)