Creating a Jira Service Desk internal comment on linked issues

On this page

Features used to implement the example

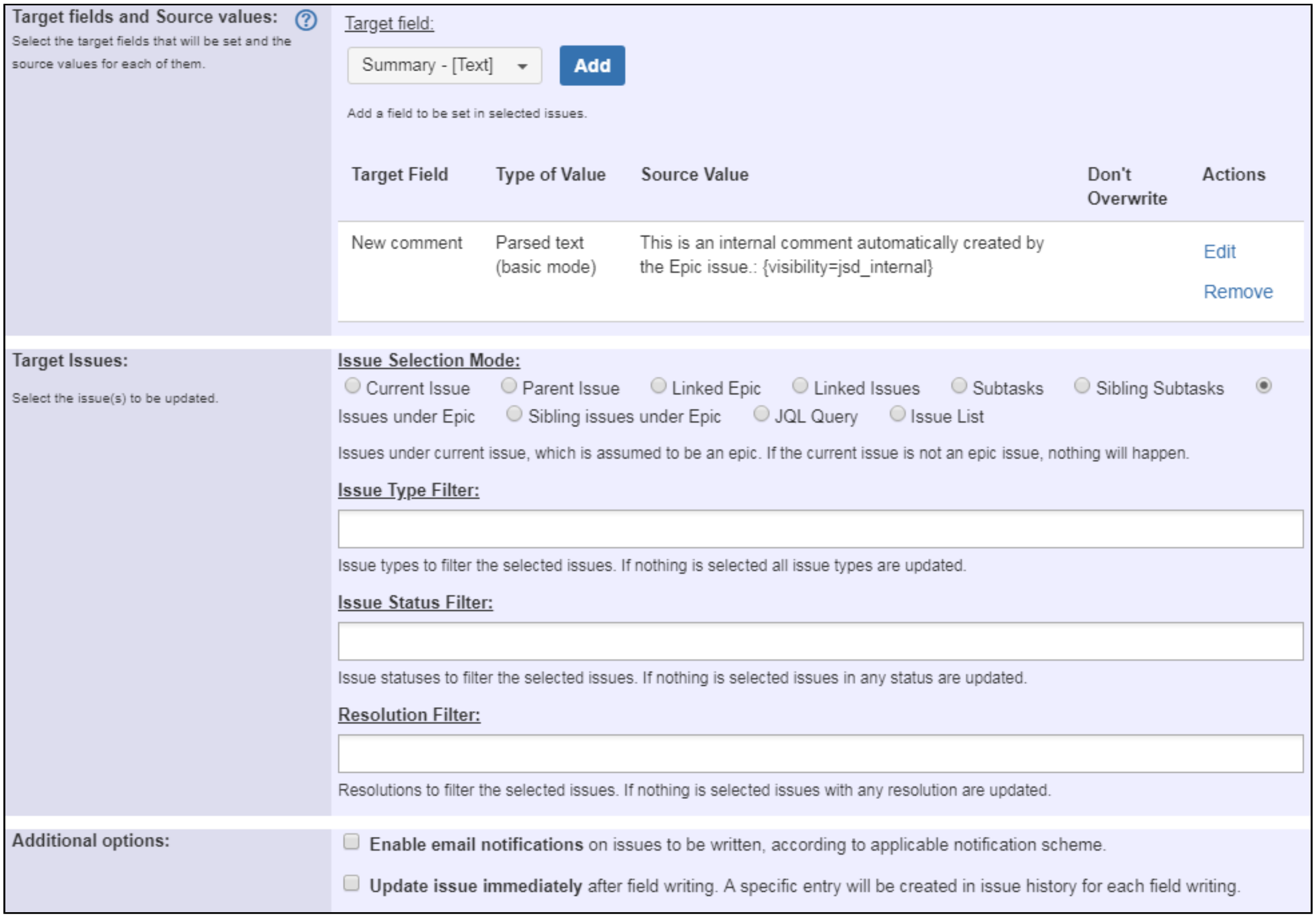
- **Update issue fields**

Example: Creating a Jira Service Desk internal comment on linked issues

Is this possible?: Writing to or reading from linked issues we need to set a new comment in a Service Desk project, but the comment has to be "Internal". At the moment, new comments are shared with the customer.

We can use post-functions **Write field on linked issues or subtasks** or **Update issue fields** selecting "**New comment**" or "**New comment (sends email notifications)**" as target field, for creating comments on linked issues, sub-tasks, or any issue selections by JQL or issue list expression. Comments created by this procedure will not have any visibility restriction, but we can limit the visibility by making our comments end in "**:{visibility=visibility_value}**", with **visibility_value** being a **project role name** a **group name**, "**jsd_internal**" or "**jsd_public**", as explained at: **Add a comment**.

Let's say we want to add an internal comment in all tasks of an Epic issue. In this case we use the post function **Update issue fields**:



Target fields and Source values: ?
Select the target fields that will be set and the source values for each of them.

Target field:
Summary - [Text] ▾ **Add**
Add a field to be set in selected issues.

Target Field	Type of Value	Source Value	Don't Overwrite	Actions
New comment	Parsed text (basic mode)	This is an internal comment automatically created by the Epic issue.: {visibility=jsd_internal}		Edit Remove

Target Issues:
Select the issue(s) to be updated.

Issue Selection Mode:
○ Current Issue ○ Parent Issue ○ Linked Epic ○ Linked Issues ○ Subtasks ○ Sibling Subtasks ● Issues under Epic ○ Sibling issues under Epic ○ JQL Query ○ Issue List

Issues under current issue, which is assumed to be an epic. If the current issue is not an epic issue, nothing will happen.

Issue Type Filter:

Issue types to filter the selected issues. If nothing is selected all issue types are updated.

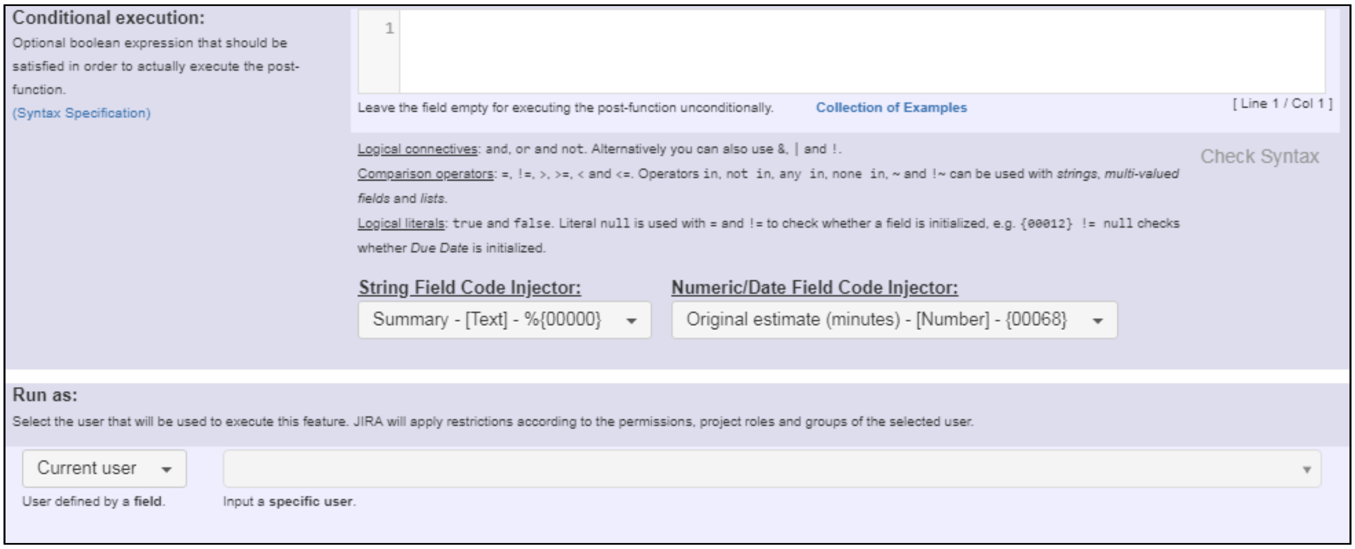
Issue Status Filter:

Issue statuses to filter the selected issues. If nothing is selected issues in any status are updated.

Resolution Filter:

Resolutions to filter the selected issues. If nothing is selected issues with any resolution are updated.

Additional options:
□ **Enable email notifications** on issues to be written, according to applicable notification scheme.
□ **Update issue immediately** after field writing. A specific entry will be created in issue history for each field writing.



Conditional execution:
Optional boolean expression that should be satisfied in order to actually execute the post-function.
(Syntax Specification)

Leave the field empty for executing the post-function unconditionally. Collection of Examples [Line 1 / Col 1]

Logical connectives: and, or and not. Alternatively you can also use &, | and !.
Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with *strings*, *multi-valued fields* and *lists*.
Logical literals: true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether *Due Date* is initialized.

Check Syntax

String Field Code Injector:
Summary - [Text] - %{00000} ▾

Numeric/Date Field Code Injector:
Original estimate (minutes) - [Number] - {00068} ▾

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.

Current user ▾
User defined by a **field**.

Input a **specific user**.

Text to be parsed is: **`This is an internal comment created by the Epic issue.: {visibility=jsd_internal}`**

Other examples of that functions

Write field on linked issues or sub-tasks

Page: Add and remove a single or a set of items from multi valued fields

Related Usage Examples

- Creating a Jira Service Desk internal comment
  - example
  - post-function

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

Write field on issues returned by JQL Query or Issue List

Page: Add and remove a single or a set of items from multi valued fields
Page: Compose dynamic text by inserting field values in a text template
Page: Creating a Jira Service Desk internal comment
Page: Creating a Jira Service Desk internal comment on linked issues
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: Parse Email adresses to watchers list
Page: Set priority for issues that have been in a certain status for longer than 24 hours
Page: Transition linked issues in currently active sprint
Page: Transition only a sub-task among several ones
Page: Using project properties to calculate custom sequence numbers
Page: Writing a comment to blocked issues when blocking issues are resolved

- Limit the number of hours a user can log per day
    - example
    - validator
    - post-function
    - work-log
- Using project properties to calculate custom sequence numbers
    - example
    - post-function
    - calculated-field
    - project-properties
- Set a date based on current date
    - example
    - post-function
- Setting the priority depending on the multiplication of custom fields
    - example
    - calculated-field
    - post-function
- Parse Email adresses to watchers list
    - example
    - post-function
- Set the assignee based on a condition
    - example
    - post-function
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
    - example
    - post-function
    - custom-field
    - sub-task
- Create a static set of sub-tasks with unique summaries
    - example
    - post-function
- Triage Jira Service Desk email requests (Move issues)
    - example
    - post-function
    - move
    - transition-issue
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)
    - example
    - post-function
    - transition
- Transition sub-tasks when parent is transitioned
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Transition only a sub-task among several ones
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
    - example
    - post-function
    - sub-task
    - transition
    - outdated