# Add and remove a single or a set of items from multivalued fields

#### On this page

- · Features used to implement the examples
- Example 1: Remove "Affects Version/s" from "Fix Version/s" in current issue
- Example 2: Add components in sub-tasks to components in parent issue
- Alternative implementation
- Example 3: Remove reporters of blocking and cloned issues from a "Multi User" field called "Team"
- Alternative implementation
- Other examples of that functions
- Related Usage Examples

### Features used to implement the examples

- · Copy parsed text to a field
- Read fields from linked issues or sub-tasks

Multi Valued fields are fields that allow you to select more than one item:

- Checkboxes
- Select List (multiple choices)
- User Picker (multiple users)
- Group Picker (multiple groups)
- Version Picker (multiple versions)
- Fixed versions
- Affected versions
- Labels
- Components
- Request Participants
- Issue Picker
- Insight Asset Management for Jira's Multi Select

You can use operators + and - to add or remove lone or set of items from Multi Valued fields.

#### Adding/Removing values from Current issue or Parent issue

We use post-function Copy parsed text to a field to compose the text with operators + and -, and copy it into a multi-valued field of current issue or parent issue (in the case of sub-tasks).

#### Adding/Removing values from other issues

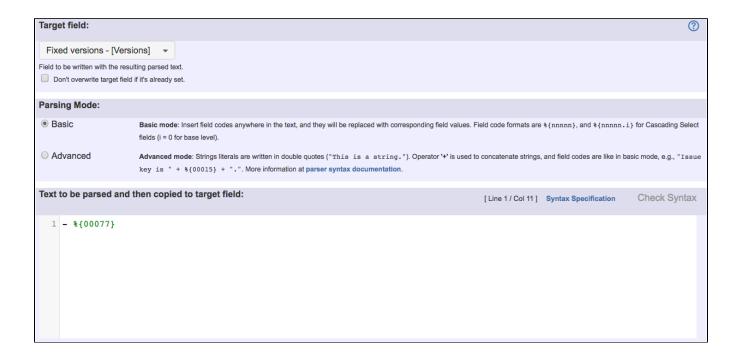
- To add or remove a single item or a set items from a multi-valued fields in other issues (linked issues, sub-tasks, or issues returned by a JQL query) different from current issue, we use any of the following post-functions:
  - Write field on linked issues or sub-tasks
  - Update issue fields

#### Adding and removing values in a single post-function

A string like + item\_1, item\_2, item\_3, - item\_4, item\_5 will add items 1, 2, 3 and remove items 4, 5. You can also inject field codes in the string, and they will be replaced by its corresponding values when parsed by post-function at run time.

# Example 1: Remove "Affects Version/s" from "Fix Version/s" in current issue

Since we are setting a field in current issue, we can do it in a single step using post-function **Copy parsed text to a field** with the following configuration:



Text to be parsed is: - %{00077}

Note that:

• %{00077} is field code for "Affects Version/s"

Once configured, the transition looks like this:

# The following will be processed after the transition occurs 1. The following text parsed in basic mode will be copied to Fixed versions: - %{Affected versions} This feature will be run as user in field Current user.

# Example 2: Add components in sub-tasks to components in parent issue

We will explain 2 different configurations to implement this same behavior.

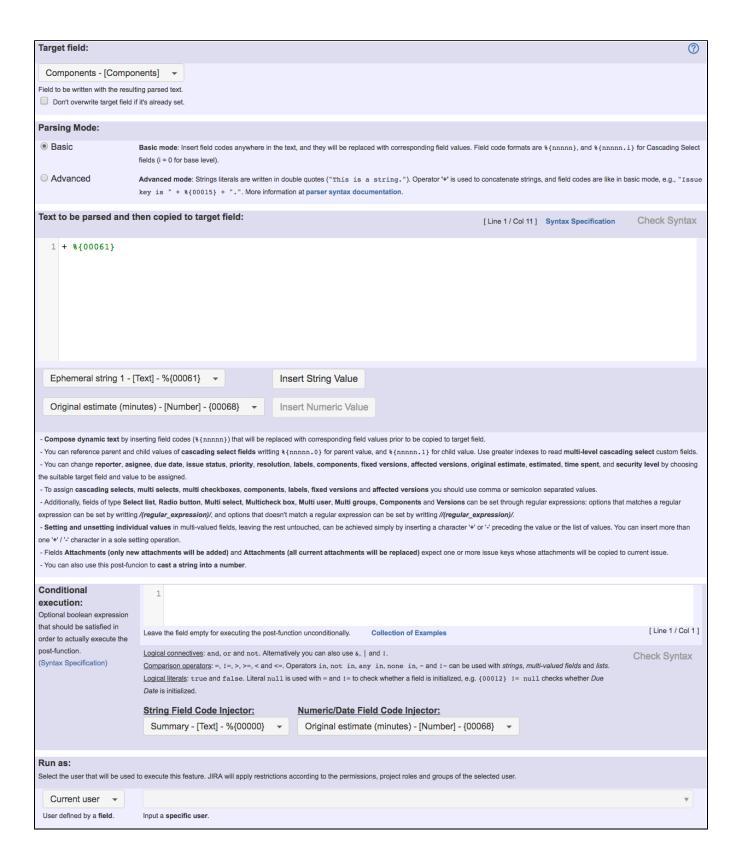
Use post-function Read fields from linked issues or sub-tasks to read Components from sub-tasks and write it into Ephemeral string 1 virtual field:

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						
	Add a field to be set in current issue.						
	Target Field	Type of Value	Source Value	Calculated Value	Don't Overwrite	Actions	
	Ephemeral string 1	Field in selected issues	Components			Edit Remove	
Filtering by issue link type:	is blocked by blocks is cloned by clones is duplicates duplicates relates to relates to	d by	ssue link types will be read.				
Read also subtasks fulfilling condition on issue type, status and project:	This option only makes	sense when current issu	re itself is not a subtask.				
Read also sibling subtasks fulfilling condition on issue type, status and project:	Sibling 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:	Car						
by toode types	□ ■ Bug						
	Epic						
	☐	ement					
	□ the New Fe	ature					
	□						
	☐ ☑ Task ☐ ☑ Sub-tas	k					
			n't select any, it won't be aplied any filter by issue	type. In that case a	III the issue types w	ill be read.	

Filtering linked issues or subtasks by status:	→ Open				
Linked issues or subtasks belong to:	<ul> <li>any project</li> <li>current project</li> <li>any but current project</li> </ul>				
Filtering by field values: Optional boolean expression that should be satisfied by linked issues and subtasks. (Syntax Specification)	Leave field empty for no filtering. [Line 1 / Col 1]				
	Logical connectives: or, and and not. Alternatively you can also use  , & and 1.  Comparison operators: =, !=, >, >=, < and <=. Operators -, !-, in, not in, any in and none in can be used with strings, multivalued 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.				
	String Field Code Injector:				
	Summary - [Text] - %{00000} 🔻				
	Field Code for Current Issue Field Code for Linked Issues / Subtasks				
	Numeric/Date Field Code Injector:  Original estimate (minutes) - [Number] - {00068}				
	Field Code for Current Issue Field Code for Linked Issues / Subtasks				
	Example 1: {00012} <= ^{00012} will require that linked issues and subtasks have Due Date equal or later than current issue's Due Date.  Example 2: %{00074} ~ ^%{00074} AND ^%{00017} in ["Blocker", "Critical"] will require that linked issues and subtasks have Fixed versions contained in current issue's Fixed versions and Priority is Blocker or Critical.				
Read linked issues and subtasks recursively:	Issues and subtasks transitively linked will also be read, provided they fulfill stated filtering conditions. Issues are read recursively without depth limit, but each selected issue is read only once.				
Read also current issue:	Current issue will be included in the issue selection, i.e., current issue's field value will also be read.				

Conditional execution: Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification)	1 Leave the field empty for executing the post-function unconditionally.  Collection of Examples	[ Line 1 / Col 1 ]	
(Cyrial openicalist)	Logical connectives: and, or and not. Alternatively you can also use s,   and !.  Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multivalued 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.		
	String Field Code Injector:  Summary - [Text] - %{00000}   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	с.	Ψ	

Use post-function Copy parsed text to a field to compose string with operator + and Components of sub-tasks read in previous step and write it into main issue's field Components:

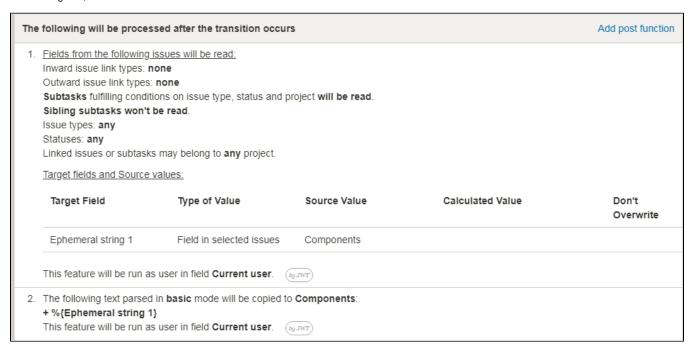


Text to be parsed is: + %{00061}

#### Note that:

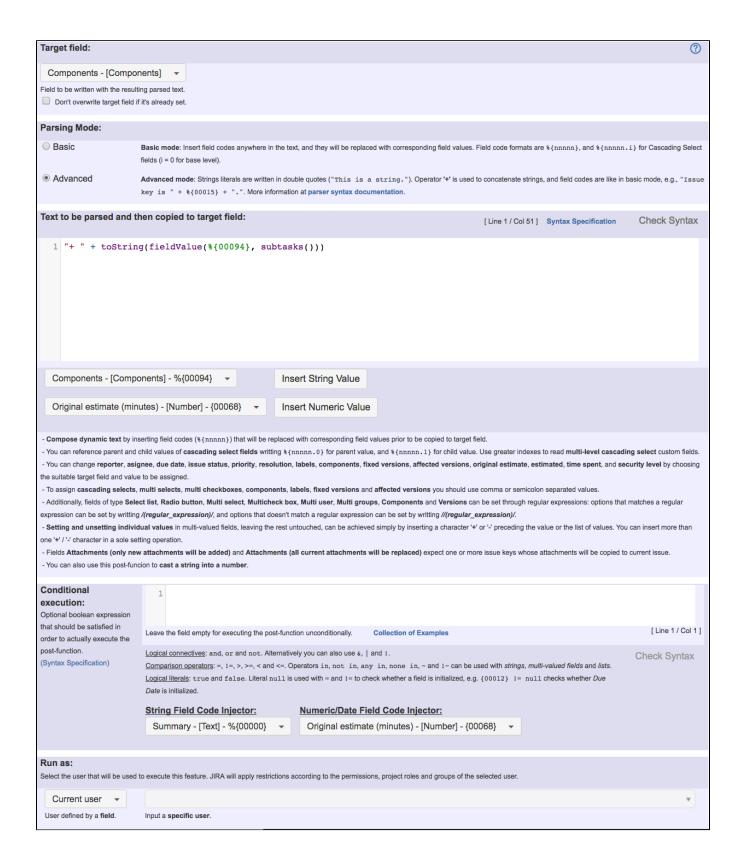
• %{00061} is field code for "Ephemeral string 1"

Once configured, the transition looks like this:



# Alternative implementation

Using one single post-function, use post-function Copy parsed text to a field with the following configuration:



Text to be parsed: "+ " + toString(fieldValue(%{00094}, subtasks()))

Once configured, the transition looks like this:

#### The following will be processed after the transition occurs

Add post function

- 1. The following text parsed in advanced mode will be copied to Components:
  - "+ " + toString(fieldValue(%{Components}, subtasks()))

This feature will be run as user in field Current user.



# Example 3: Remove reporters of blocking and cloned issues from a "Multi User field called "Team"

We will explain 2 different configurations to implement this same behavior:

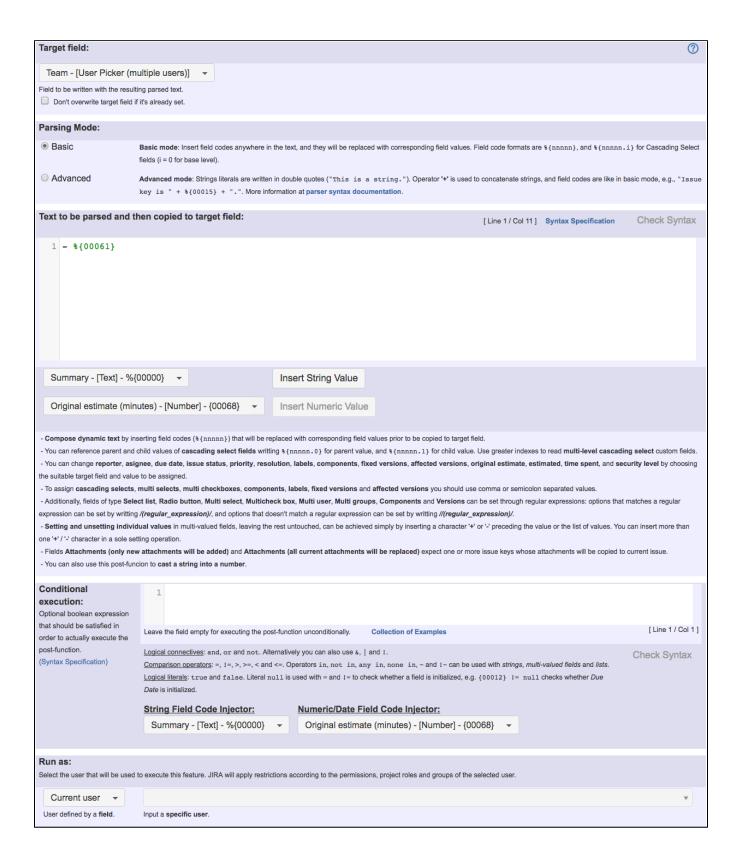
Use post-function Read fields from linked issues or sub-tasks to read field Reporter from linked issues and write it into Ephemeral string 1:

Target fields and Source values:  Select the target fields that will be set and	Target field:						
the source values for each of them.	Summary - [Text] ▼ Add						
	Add a field to be set in current issue.						
	Target Field	Type of Value	Source Value	Calculated Value	Don't Overwrite	Actions	
	Ephemeral string 1	Field in selected issues	Reporter			Edit Remove	
Filtering by issue link type:	✓ is blocked b	y					
	blocks						
	✓ is cloned by						
	clones						
	is duplicated by						
	duplicates						
	relates to						
	relates to						
	Only issues linked to cur	rrent issue by selected is	ssue link types will be read.				
Read also subtasks fulfilling condition on issue type, status and project:	This option only makes	sense when current issu	ie itself is not a subtask.				
Read also sibling subtasks fulfilling condition on issue type, status and project:	Sibling 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:	Car						
-,	□ □ Bug						
	Epic						
	☐ Improve	ement					
	□ <b>±</b> New Fe	ature					
	Story						
	☐ ✓ Task ☐ Sub-tas	k					
			n't select any, it won't be aplied any filter by issue	type. In that case a	II the issue types w	ill be read.	
	- Secret Save types will		apinco uny micr dy 155uc	-, 0000 0			

Filtering linked issues or subtasks by status:	Open  In Progress  Reopened  Resolved  To Do  Acceptance  Fail  Pass  Retest  Active			
	Selected statuses will be read, but if you don't select any, it won't be aplied any filter by status. In that case issues in any status will be read.			
Linked issues or subtasks belong to:	<ul> <li>any project</li> <li>current project</li> <li>any but current project</li> </ul>			
Filtering by field values: Optional boolean expression that should be satisfied by linked issues and subtasks. (Syntax Specification)	Leave field empty for no filtering. [Line 1 / Col 1]			
	Logical connectives: or, and and not. Alternatively you can also use  , & and 1.  Comparison operators: =, 1 =, >, >=, < and <=. Operators -, 1 -, in, not in, any in and none in can be used with strings, multivalued fields and lists.  Logical literals: true and false. Literal null is used with = and 1 = to check whether a field is initialized, e.g. {00012} != null checks whether Due Date is initialized.			
	String Field Code Injector:			
	Summary - [Text] - %{00000} 🔻			
	Field Code for Current Issue Field Code for Linked Issues / Subtasks			
	Numeric/Date Field Code Injector:  Original estimate (minutes) - [Number] - {00068}			
	Field Code for Current Issue Field Code for Linked Issues / Subtasks			
	Example 1: {00012} <= ^{00012} will require that linked issues and subtasks have Due Date equal or later than current issue's Due Date.  Example 2: \{00074\} ~ ^\{00074\} AND ^\{00017\} in ["Blocker", "Critical"] will require that linked issues and subtasks have Fixed versions contained in current issue's Fixed versions and Priority is Blocker or Critical.			
	Issues and subtasks transitively linked will also be read, provided they fulfill stated filtering conditions. Issues are read recursively without depth limit, but each selected issue is read only once.			
Read also current issue:	Current issue will be included in the issue selection, i.e., current issue's field value will also be read.			

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 ε,   and !.  Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multivalued 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.  String Field Code Injector:  Numeric/Date Field Code Injector:		
	Summary - [Text] - %{00000}   Original estimate (minutes) - [Number] - {00068}		
Run as:			
	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			

Use post-function Copy parsed text to a field to compose string with operator – and Reporter previously read in step 1 and write it into Multi User field Team:

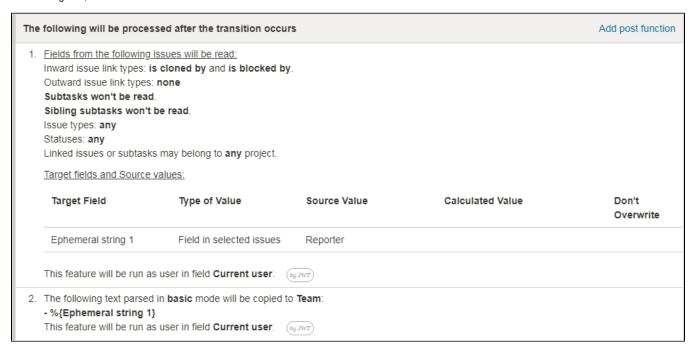


Text to be parsed is: - %{00061}

#### Note that:

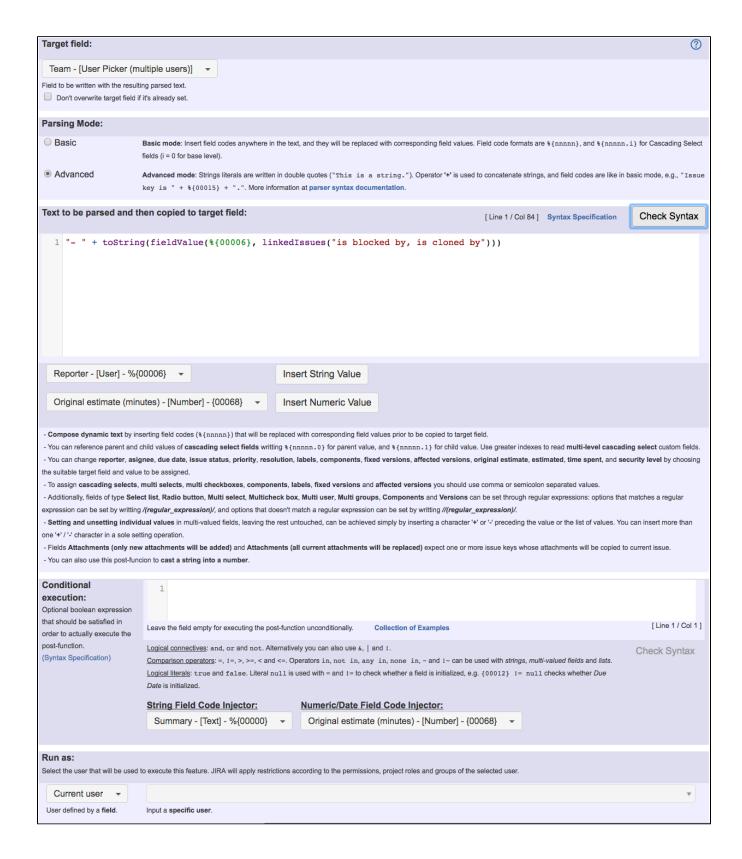
• %{00061} is field code for "Ephemeral string 1"

Once configured, the transition looks like this:



# Alternative implementation

Using one single post-function,use post-function Copy parsed text to a field with the following configuration:



Text to be parsed: "- " + toString(fieldValue(%{00006}, linkedIssues("is blocked by, is cloned by")))

Once configured, the transition looks like this:

- The following text parsed in advanced mode will be copied to Team:
  - "- " + toString(fieldValue(%{Reporter}, linkedIssues("is blocked by, is cloned by")))

This feature will be run as user in field Current user.



## 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

# Copy parsed text to a field

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: Add current user to comment

Page: Add or remove request participants

Page: Add watchers from a part of the issue summary: "Summary\_text watcher1, watcher2, watcher3, ...

Page: Assign issue based on the value of a Cascading Select custom

Page: Assign issue to last user who executed a certain transition in the

Page: Automatically close resolved sub-tasks when parent issue is

Page: Automatically reopen parent issue when one of its sub-tasks is

reopened

Page: Calculate the time elapsed between 2 transition executions

Page: Close parent issue when all sub-tasks are closed Page: Combine the values of several Multi-User picker fields

Page: Compose a parsed text including the "full name" or a user selected in a User Picker custom field

Page: Compose dynamic text by inserting field values in a text template

Page: Copy issue labels to a custom field

Page: Copy the value of a user property into a user picker

Page: Create a comment in sub-tasks when parent transitions

Page: Execute transition in epic

Page: Getting the number of selected values in a custom field of type Multi Select

Page: Limit the number of hours a user can log per day

Page: Make a sub-task's status match parent issue's current status on

Page: Make parent issue progress through its workflow

Page: Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"

Page: Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status

Page: Parse Email adresses to watchers list

Page: Parsing text from last comment and appending it to issue's

Page: Remove versions selected in a version picker custom field

Page: Replace certain issue link types with different ones

Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status

Page: Set a Select or Multi-Select field using regular expression to express the values to be assigned

Page: Set assignee depending on issue type

Page: Set field depending on time passed since issue creation

## Related Usage Examples

- Add and remove a single or a set of items from multi valued fields
  - o example
  - o post-function
  - o custom-field
  - o issue-links
  - o sub-task
- Prevent transitioning when there is a blocking issue
  - o example
  - validator
  - o issue-links
  - o sub-task
  - o transition
- Make linked issues, sub-tasks and JQL selected issues progress through its workflows
  - example
  - o condition
  - o validator
  - o post-function o issue-links
  - o sub-task
  - o transition
- · Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC'
  - example
  - o post-function
  - o issue-links
  - o sub-task
  - o work-log
- Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
  - o example
  - o post-function
  - o issue-links
  - o sub-task
  - o work-log
- Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
  - o example
  - o post-function
  - o custom-field
  - o issue-links
- · Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
  - example
  - o post-function
  - o custom-field
  - o sub-task
- · Add all assignees of certain sub-task types to a "Multi-User Picker" custom field
  - o example
  - post-function
  - o custom-field
  - o sub-task
- Update Cascading Select custom field with a value of the field in parent issue
  - o example
  - o post-function
  - o custom-field
  - o sub-task
- · Create a sub-task for each user selected in a Multi-User Picker
  - o example
  - o post-function

Page: Set priority for issues that have been in a certain status for longer than 24 hours

Page: Set security level based on groups and project roles the reporter or creator are in

Page: Transition linked issues in currently active sprint Page: Transition only a sub-task among several ones

Page: Transition parent issue only when certain issue sub-task types

Page: Update Cascading Select custom field with a value of the field in parent issue

Page: Update checkboxes custom field if a file has been attached during a transition

Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments

Page: Writing a comment to blocked issues when blocking issues are

resolved

- o custom-field
- o sub-task
- Validate only issue links created in transition screen
  - example
  - o validator
  - o issue-links
- Require issue link when resolving as duplicate
  - o example
  - o validator
  - o issue-links
- Ensure that all issues linked with a certain issue link type have "Due Date" field set
  - o example
  - o validator
  - o issue-links
- Block an epic's transition depending on linked issues status and due date
  - o example
  - o validator
  - o issue-links
  - o transition
- · Writing a comment to blocked issues when blocking issues are resolved
  - o example
  - o post-function
  - o issue-links