

# Transition issue

This post function automatically executes a **specific transition** on the **selected issues** or transitions issues **to a selected status**.

It is the preferred way of executing transitions. However, in some cases, e.g. the automatic transitioning of newly created issues using the [Create issue](#) post function, users might want to use the following alternative field codes:

- [Issue status](#)
- [Issue status \(delayed writing\)](#)
- [Execute transition](#)
- [Execute transition \(delayed execution\)](#)



## Configuration

### Target issue

Select the issue(s) to be transitioned. The following options are available:

Option	Description
Current issue	The <b>current issue</b> will be transitioned.
Parent issue	The current issue's <b>parent</b> will be transitioned.
Sub-tasks	The current <b>issue's sub-tasks</b> will be transitioned. If you choose this option you can <b>further filter</b> your <b>selection</b> (e.g. by issue type or status). <b>Only useful</b> for <b>standard issues</b> .
Sibling sub-tasks	The current issue's sibling sub-tasks will be transitioned. <b>Only useful</b> if the current issue is a <b>sub-task</b> .
Linked issues	<b>Linked issues</b> will be transitioned. If you choose this option you can <b>further filter</b> your <b>selection</b> (e.g. by issue type or status).
Linked epic	The related <b>epic</b> will be transitioned.
Issues under epic	<b>All issues</b> related to the same epic will be transitioned. If you choose this option you can <b>further filter</b> your <b>selection</b> (e.g. by issue type or status).
Sibling issues under epic	<b>All sibling issues</b> related to the same epic will be transitioned. If you choose this option you can <b>further filter</b> your <b>selection</b> (e.g. by issue type or status).
Set target issue manually (parser expression)	Define the issues that will be transitioned by defining an <b>expression</b> . The <a href="#">JQL mode</a> and <a href="#">Issue list mode</a> are available.  The expression must return either a valid <b>JQL expression</b> or an issue list with <b>issue keys</b> .

### Mode

Select the **transition** to be executed or the **status** to be transitioned to. The following options are available:

Option	Description
--------	-------------

Execute transition	<p>Either select an existing <b>transition</b> from any available workflow or <b>set the transition manually</b> using a parser expression in <a href="#">Basic text mode</a>.</p> <p>The expression must return a <b>transition name</b>.</p>
Transition to status	<p>Either select an existing <b>status</b> from any available workflow or set the status manually using a parser expression in <a href="#">Basic text mode</a>.</p> <p>The expression must return a <b>status name</b>.</p> <p>If <b>more than one</b> transition is <b>available</b> to reach the configured status, the <b>first</b> transition (found by Jira) will be executed.</p>

Both, **transition** or target **status**, **have to be available** from the issues' current status. The provided list, however, contains **all** statuses and transitions in the system.

## Additional options

Choose between the following options (multi choice is possible):

Option	Description
Ignore conditions	Existing conditions in the transitions to be executed will be ignored.
Ignore validators	Existing validators in the transitions to be executed will be ignored.
Ignore project permissions	The "Transition Issues" project permission will be ignored for the user running this post function.
Delayed execution	<p>The execution of this post function will be delayed by the value specified in <b>milliseconds</b>.</p> <p>This parameter is useful, when multiple operations have to be performed in a single transition. The delayed execution ensures that all <b>previous operations have been completed</b>, e.g. field values that are referenced in this post function are present.</p> <p>The maximum delay is 60,000 milliseconds (60 seconds).</p>

## Conditional execution

You can **optionally** specify a [logical expression](#) to define the circumstances (or conditions) under which the post function should be executed.

The result of the logical expression must return a boolean value of either:

- **true** the post function will be executed
- **false** the post function will **not** be executed

Using the **conditional operator**, even complex or multi-layered conditions can be constructed.

Make sure to learn more about defining logical expressions and browse through the various **examples** here: [Logical mode](#)

## Run as

Select which **user** will be used to execute this post function. By default this parameter is set to the **current user**. You can also use field codes to run the function as a dynamic user (e.g. current assignee).

Make sure that the user running the post function has all the **relevant permissions** to perform the actions defined in the configuration (e.g. "Update Issues")!

If you want to keep track the actions being performed automatically, we suggest to create a **dedicated JWT account**, granted all relevant **permissions**, and use it in the Run as parameter to identify which changes have been made with JWT.



## Use cases and examples

Use case	JWT feature	Workflow function	Parser functions	Label
Fast-track transition issues assigned to the project lead		Transition issue		STAFF PICK
Keep the status of an issue and its linked issues in sync		Transition issue		
Keep the status of parents and sub-tasks in sync (post function use case)		Transition issue		
Reopen parent issue, if a sub-task is reopened		Transition issue		
Start Progress on parent, if sub-tasks are started		Transition issue		STAFF PICK

If you still have questions, feel free to refer to our [support team](#).