# Transition affected issues when releasing a version (manually)

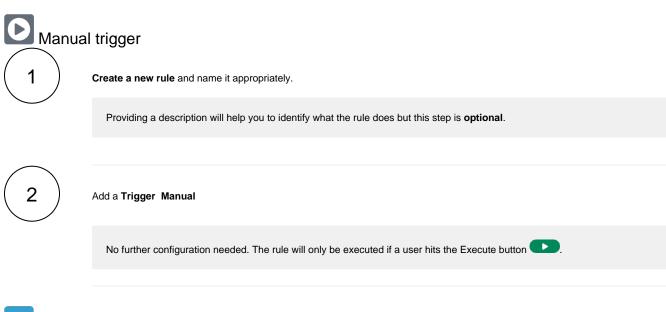
## On this page

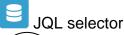
Manual trigger | JQL selector | Transition issue action | Screencast | Import the example | Related use cases

#### Use case

During new software releases you get to the point where all related issues within Jira have to be **releas ed**. You could do this using a **project event trigger** as described in **this use case**. If you want to have a more fine-grained control you might want to kick-off this process manually. At this point our **Manual Trigger** comes in handy!

The following rule will build on the project event use case but instead use the manual trigger.





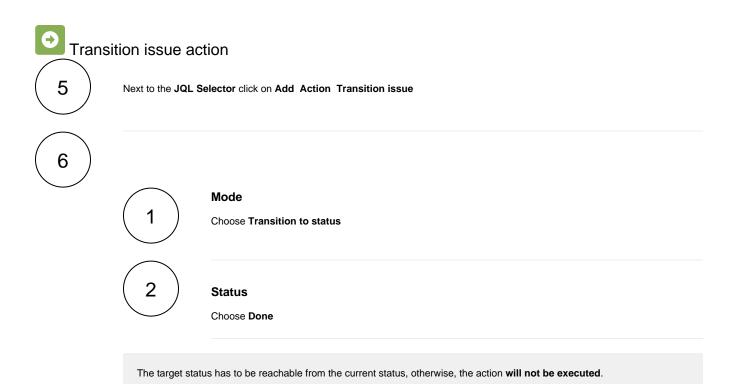
Add Selector JQL Selector



### **JQL Query**

Use the following Expression:

fixversion = nnnnn # Replace nnnnn with the actual version that got released



# Screencast

This is how the configuration above should look on your screen

Your browser does not support the HTML5 video element



# Import the example

Import the JSON file below to get started in no time.

# **JSON**

After importing the JSON file, make sure to **check** the **configuration** of the rule. Non-existing configuration elements (issue types, fields, values etc.) will be highlighted.

```
"name": "Transition affected issues when releasing a version",
   "description": "",
    "creator": "admin",
    "status": false,
    "triggerData": "",
    "triggerType": "MANUAL",
    "configuration": {
        "refs": [
           "issue",
           "system",
           "trigger.issue",
           "trigger.parent"
        "actingUser": "field_00020",
        "triggerType": ""
   },
    "children": [
       {
            "sequence": 0,
            "type": "JQL_SELECTOR",
            "ruleEntityType": "SELECTOR",
            "configuration": {
                "refs": [
                    "system"
                "jql": "project = TIS and fixVersion = 1.9",
                "jqlParsingMode": "jql",
                "actingUser": "field_00020"
           },
            "children": [
                {
                    "sequence": 0,
                    "type": "TRANSITION_ISSUE",
                    "ruleEntityType": "ACTION",
                    "configuration": {
                        "refs": [
                           "issue",
                            "issues",
                            "selector.issue",
                            "selector.parent",
                            "system"
                        ],
                        "option": "status",
                        "status": "10012",
                        "actingUser": "field_00020"
                    },
                    "children": null,
                    "hasChildren": false
            "hasChildren": true
   ],
    "hasChildren": true
}
```



## Related use cases

Title Automated action JWT feature Label

Transition affected issues when releasing a version (manually)	Transition issue action	<b>\$</b>	
Close epic when stories are done	Transition issue action	<b>\$</b>	
Automatically close parent when all sub-tasks are done	Transition issue action	<b>\$</b>	STAFF PICK
Resolve issues when all options of a checklist are checked	Transition issue action	<b>\$</b>	
Schedule issues for re-submission	Transition issue action	<b>\$</b>	
Schedule the escalation of overdue issues	Transition Issue	<b>\$</b>	
Keep the status of parents and sub-tasks in sync	Transition issue action	<b>\$</b>	STAFF PICK
Transition affected issues when releasing a version (automatically)	Transition issue action	<b>\$</b>	STAFF PICK
Transition issues when assigned to a developer	Transition issue action	<b>\$</b>	
Automatically close sub-tasks when parent is completed	Transition issue action	<b>\$</b>	STAFF PICK
Re-open issue when a new comment is added	Transition issue action	<b>\$</b>	
Close stories when epic is done	Transition issue action	<b>&amp;</b>	

If you still have questions, feel free to refer to our support team.