Keep the status of parents and sub-tasks in sync

On this page

Use case

Keeping parent issues and sub-tasks in sync, as described in **this use case**, is a lot of manual work. This automation rule will update the status of all sub-tasks based on the **parent status**: When the parent issue is rejected, all sub-tasks are also being rejected.

Should I use an automation rule or a workflow post function?

When to use this automation rule and when to use a workflow post function?

This use case is very useful, if there are several software projects with different workflows where all related issues have to be closed on releasing a new version. Configuring just one automation rule might then be the easier way to go. But if this scenario is only necessary for one workflow, the **Transition Issues post function** would make more sense. In this case you might want to check out our **corresponding post function use case**.

Issue transitioned event

1 **Create a new rule** and name it appropriately.

Providing a description will help you to identify what the rule does but this step is **optional**.

2 Add a **Trigger Issue transitioned event**

No further configuration needed. The automation rule is triggered every time an issue is being transitioned.

Boolean condition

3 Add a **Condition Boolean Condition**

4 **Expression**

Use the following **Parser Expression**:

```
%{trigger.issue.status} = "Rejected"
```

more info...

Boolean expressions are **logical constructions** that return **`true`** or **`false`** .
In this use case, the expression will only return true if the status of the **issue triggering the rule** is REJECTED.

Issue selector

5 Next to the **Boolean Condition** click on **Add Selector Issue Selector**

6 **Target Issue(s)**

Choose **Sub-tasks**

Transition issue action

7 Next to the **Issue Selector** click on **Add Action Transition issue**

1 **Mode**

Transition to status

2 **Status**

Choose **Rejected**

8 Enable the rule by clicking on the **Enable button** 

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": "Keep the status of parents and sub-tasks in sync",
    "description": "",
    "creator": "admin",
    "status": false,
    "triggerData": "",
    "triggerType": "ISSUE_TRANSITIONED_EVENT",
    "configuration": {
        "refs": [
            "issue",
            "system",
            "trigger.issue",
            "trigger.parent"
        ],
        "triggerType": ""
    },
    "children": [
        {
            "sequence": 1,
            "type": "BOOLEAN_CONDITION",
            "ruleEntityType": "CONDITION",
            "configuration": {
                "refs": [
                    "issue",
                    "project",
                    "system",
                    "trigger",
                    "trigger.issue",
                    "trigger.parent"
                ],
                "expression": "%{trigger.issue.status} = \"Rejected\"",
                "expressionParsingMode": "logical",
                "actingUser": "field_00020"
            },
            "children": [
                {
                    "sequence": 0,
                    "type": "ISSUE_SELECTOR",
                    "ruleEntityType": "SELECTOR",
                    "configuration": {
                        "refs": [
                            "issue",
                            "project",
                            "system",
                            "trigger",
                            "trigger.issue",
                            "trigger.parent"
                        ],
                        "option": "subtasks",
                        "issueListExpressionParsingMode": "issues",
                        "actingUser": "field_00020"
                    },
                    "children": [
                        {
```

```
                            "sequence": 0,
                            "type": "TRANSITION_ISSUE",
                            "ruleEntityType": "ACTION",
                            "configuration": {
                                "refs": [
                                    "issue",
                                    "issues",
                                    "project",
                                    "selector.issue",
                                    "selector.parent",
                                    "system",
                                    "trigger",
                                    "trigger.issue",
                                    "trigger.parent"
                                ],
                                "option": "status",
                                "status": "10002",
                                "actingUser": "field_00020"
                            },
                            "children": null,
                            "hasChildren": false
                        }
                    ],
                    "hasChildren": true
                }
            ],
            "hasChildren": true
        }
    ],
    "hasChildren": true
}
```

Related use cases

Title	Automated action	JWT feature	Label
Automatically close parent when all sub-tasks are done	Transition issue action		STAFF PICK
Automatically close sub-tasks when parent is completed	Transition issue action		STAFF PICK
Close epic when stories are done	Transition issue action		
Close stories when epic is done	Transition issue action		
Keep the status of parents and sub-tasks in sync	Transition issue action		STAFF PICK
Re-open issue when a new comment is added	Transition issue action		

Resolve issues when all options of a checklist are checked	Transition issue action		
Schedule issues for re-submission	Transition issue action		
Schedule the escalation of overdue issues	Transition Issue		
Transition affected issues when releasing a version (automatically)	Transition issue action		STAFF PICK
Transition affected issues when releasing a version (manually)	Transition issue action		
Transition issues when assigned to a developer	Transition issue action		

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