

Automatically close parent when all sub-tasks are done

On this page

[Issue transitioned event](#) | [JQL condition](#) | [Boolean condition](#) | [Issue selector](#) | [Transition issue](#) | [Screencast](#) | [Import the example](#) | [Related examples](#)

Use case

Avoid the manual work by closing the parent after checking if all sub-tasks are done, let automation do it for you!



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**



JQL condition

3

Add the following expression:

```
issuetype in subtaskIssueTypes()
```



Boolean condition

4

Next to the **JQL condition** click on **Add Boolean condition**

1

Add the following expression:

```
%{trigger.issue.status} = "Done" and count(siblingSubtasks()) = count  
(filterByStatus(siblingSubtasks(), "Done"))
```

This way the rule execution will only continue, if the current issue was transitioned to **Done** and if all other sub-tasks under the current parent issue are also in status **Done**.



Issue selector

5

Next to the **Boolean condition** click on **Add Issue selector**

1

Choose **Parent issue** as the target issue.



Transition issue

6

Finally, next to the **Issue selector** click on **Add Action Transition issue**

1

Set the mode to **Transition to status** and choose the status **Done**.

7

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



Screencast

No screencast available at the moment.



Import the example

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

JSON

```
{
  "name": "Automatically close parent when all sub-tasks are done",
  "description": "",
  "creator": "admin",
  "status": false,
  "triggerData": "",
  "triggerType": "ISSUE_TRANSITIONED_EVENT",
  "configuration": {
    "refs": [
      "issue",
      "system",
      "trigger.issue",
      "trigger.parent"
    ],
  },
}
```

```

    "triggerType": ""
  },
  "children": [
    {
      "sequence": 0,
      "type": "JQL_CONDITION",
      "ruleEntityType": "CONDITION",
      "configuration": {
        "refs": [
          "issue",
          "project",
          "system",
          "trigger",
          "trigger.issue",
          "trigger.parent"
        ],
        "jql": "issuetype in subtaskIssueTypes()",
        "jqlParsingMode": "jql",
        "actingUser": "field_00020"
      }
    },
    {
      "sequence": 0,
      "type": "BOOLEAN_CONDITION",
      "ruleEntityType": "CONDITION",
      "configuration": {
        "refs": [
          "issue",
          "project",
          "system",
          "trigger",
          "trigger.issue",
          "trigger.parent"
        ],
        "expression": "%{trigger.issue.status} = \"Done\" and count(siblingSubtasks()) = count(filterByStatus(siblingSubtasks(), \"Done\"))",
        "expressionParsingMode": "logical",
        "actingUser": "field_00020"
      }
    },
    {
      "sequence": 0,
      "type": "ISSUE_SELECTOR",
      "ruleEntityType": "SELECTOR",
      "configuration": {
        "refs": [
          "issue",
          "project",
          "system",
          "trigger",
          "trigger.issue",
          "trigger.parent"
        ],
        "option": "parent",
        "issueListExpressionParsingMode": "issues",
        "actingUser": "field_00020"
      }
    },
    {
      "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": "10012",
      "actingUser": "field_00020"
    },
    "children": null,
    "hasChildren": false
  }
],
"hasChildren": true
}
],
"hasChildren": true
}
],
"hasChildren": true
}
],
"hasChildren": true
}
}

```



Related examples

Title	Automated action	JWT feature	Label
Transition affected issues when releasing a version (manually)	Transition issue action		
Close epic when stories are done	Transition issue action		
Automatically close parent when all sub-tasks are done	Transition issue action		STAFF PICK
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		
Keep the status of parents and sub-tasks in sync	Transition issue action		STAFF PICK
Transition affected issues when releasing a version (automatically)	Transition issue action		STAFF PICK

Transition issues when assigned to a developer	Transition issue action		
Automatically close sub-tasks when parent is completed	Transition issue action		STAFF PICK
Re-open issue when a new comment is added	Transition issue action		
Close stories when epic is done	Transition issue action		

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