# How to Move Issues Automatically in Jira On-premise

Last Modified on 01/28/2026 10:30 am EST

This article shows how to move issues automatically with the External scripts library. This script handles the situation when you move issues from one workflow to another workflow on the source system.

**Pre-requisites**

| | Source side | Destination side |
|---|---|---|
| **projects** | ProjectKey_A1 <br> ProjectKey_A2 | ProjectKey_B1 <br> ProjectKey_B2 |
| **issue types** | IssueType_A1 <br> IssueType_A2 | IssueType_B1 <br> IssueType_B2 |
| **workflows** | | LocalWorkFlow1 <br> LocalWorkFlow2 |

## Configuration

## 1. Upload the File with the External Script

You need to upload the Move.groovy file into the **scripts** directory.

> External scripts library location
>
> Jira Server: **$JIRA_HOME/scripts**

## 2. Call the Script from the Sync Rules

Once all external .groovy files have been uploaded to the proper directory, you need to call the script from the Sync Rules.

> Every .groovy file includes comments with a detailed explanation on how to use it. For more

## Source Side

**Outgoing sync**

```
replica.type = issue.type
replica.project = issue.project
replica.status = issue.status
```

## Destination Side

**Incoming sync**

## Move Issues automatically according to the Specified Mapping

The method helps to move issues automatically in case the source issue has been moved to another project or the issue type has been changed on the source side.

**Incoming sync**

```
Move.receive(
 //projectMapping:
 ["ProjectKey_A1":"ProjectKey_B1", "ProjectKey_A2":"ProjectKey_B2"],
 //issueTypeMapping:
 ["IssueType_A1":"IssueType_B1", "IssueType_A2":"IssueType_B2"],
 //moveStatusMapping:
 ["LocalWorkFlow1" : ["LocalWorkFlow2" : ["OPEN": "TO DO"] ] ],
 //sameStatusIfNotInMapping
 true,
 replica, issue
 )
```

## Move Issue with Identical Workflows for all Issue Types

If the workflows are identical for all issue types, you don't need to specify the status mapping. But you still need to specify project and issue type mappings if the projects and issue types are different.

```
Move.receive(
  //projectMapping:
  ["ProjectKey_A1":"ProjectKey_B1", "ProjectKey_A2":"ProjectKey_B2"],
  //issueTypeMapping:
  ["IssueType_A1":"IssueType_B1", "IssueType_A2":"IssueType_B2"],
  // moveStatusMapping
    [:],
    true, // use the same status if mapping is not specified
  replica,
  issue
)
```

## Move Issue with Different Workflows for all Issue Types

*moveStatusMapping* - defines, the status of the issue moved to another workflow.

If an issue which is under sync is moved from LocalWorkFlow1 to LocalWorkFlow2 and the current status is OPEN, set the status TO DO.

**Example:** [...,"LocalWorkFlow1":[...,"LocalWorkFlow2":[...,"OPEN":"TO DO",...],...],...]

**Incoming sync for existing issues(change processor)**

```
if (replica.type.name != issue.type.name) {
  Move.move(
    //moveStatusMapping:
    ["LocalWorkFlow1" : ["LocalWorkFlow2" : ["OPEN": "TO DO"] ] ],
    //sameStatusIfNotInMapping
  true,
  issue,
  issue.project.key, // the key of the project where the issue is moved to here
  replica.type.name // the name of the issue type where the issue is moved to
  )
}
```

## Sync Issues Automatically Using Exalate API in Jira On-premise

This article shows how to start issue synchronization automatically with the help of Exalate API.

Sync an issue according to the specified issue key

```
def exaCl = com.atlassian.jira.component.ComponentAccessor
      .pluginAccessor
      .getEnabledPlugin("com.exalate.jiranode")
      .getClassLoader()
def exaSyncServiceClass = exaCl.loadClass(
    "com.exalate.api.trigger.ISyncInitiationService"
)
def syncService = com.atlassian.jira.component.ComponentAccessor.getOSGiComponentInstanceOfType(
      exaSyncServiceClass
)
syncService.pair(
    "PROJ-23", // specify the issue key you want
    "ConnectionTest" // set the relevant connection name
```

**Resources**

Subscribe for a weekly Exalate hack  ⬚

Academy ⬚

Blog ⬚

YouTube Channel ⬚

Ebooks ⬚

**Still need help?**

Join our Community ⬚

Visit our Service Desk ⬚

Find a Partner ⬚