

# Jira Cloud: Automatically Unexalate issues moved to another project

Last Modified on 02/28/2024 6:56 am EST

This article shows how to stop the sync automatically once the issue has been moved to another project.

## Introduction

The synchronization is set up to sync issues from the specific project automatically.

Now, if the issue has been moved from the project under sync to another project which has a different workflow, the synchronization should be stopped automatically.

*Consideration:* If the issue is synced with one or more connections, the following approach will unexalate the issue from all connections.

You can configure the synchronization behavior in different ways to achieve this:

- Jira on-premise
  - Continue sync, but don't apply any changes if the issue has been moved to another project
  - Stop the sync if the issue has been moved to another project
- Jira Cloud
  - Continue sync, but don't apply any changes if the issue has been moved to another project
  - Stop the sync if the issue has been moved to another project

## Jira on-premise

Continue the sync, but don't apply any changes if the issue has been moved to another project

Use this approach if moved issues might be moved back at some point of time.

### Source side

#### Incoming sync

```
if (!firstSync && issue.project.key != "HELP") {  
  // don't apply any changes if the local issue has been moved from the HELP project  
  return  
}  
// the rest of the script remains untouched and goes after this `if`  
// ...
```

### Destination side

## Incoming sync

```
if (firstSync && replica.project.key != "HELP") {
  return
} else
if (!firstSync && replica.project.key != "HELP") {
  // don't apply any changes if the remote issue has been moved from the HELP project
  return
}
// the rest of the script remains untouched and goes after this `if`
```

## Stop the sync if the issue has been moved to another project

Use this approach if moved issues would not be moved back at some point of time. If the issue is moved back, it triggers sync once again by creating a new twin issue.

## Source side

### Incoming sync

```
if (!firstSync && issue.project.key != "HELP") {
  // stop sync if the local issue has been moved from the HELP project
  final def _syncInitService = com.atlassian.jira.component.ComponentAccessor
    .getOSGiComponentInstanceOfType(com.exalate.api.trigger.ISyncInitiationService.class)
  _syncInitService.unexalate(issueKey.id as String)
  return
}
// the rest of the script remains untouched and goes after this `if`
```

## Destination side

### Incoming sync

```
if (firstSync && replica.project.key != "HELP") {
  return
} else
if (!firstSync && replica.project.key != "HELP") {
  // stop sync if the local issue has been moved from the HELP project
  final def _syncInitService = com.atlassian.jira.component.ComponentAccessor
    .getOSGiComponentInstanceOfType(com.exalate.api.trigger.ISyncInitiationService.class)
  _syncInitService.unexalate(issueKey.id as String)
  return
}
// the rest of the script remains untouched and goes after this `if`
```

## Jira Cloud

Continue the sync, but don't apply any changes if the issue has been moved to another project

Use this approach if moved issues might be moved back at some point of time.

## Source side

### Incoming sync

```
1 if (!firstSync && issue.project.key != "HELP") {
2   // don't apply any changes if the local issue has been moved from the HELP project
3   return
4 }
5 // the rest of the script remains untouched and goes after this `if`
```

## Destination side

### Incoming sync

```
1 if (firstSync && replica.project.key != "HELP") {
2   return
3 } else
4 if (!firstSync && replica.project.key != "HELP") {
5   // don't apply any changes if the remote issue has been moved from the HELP project
6   return
7 }
8 // the rest of the script remains untouched and goes after this `if`
```

## Stop the sync if the issue has been moved to another project

Use this approach if moved issues would not be moved back at some point of time.

If the issue is moved back, it triggers sync once again by creating a new twin issue.

## Source side

### Incoming sync

```
1 if (issue.project.key != "HELP") {
2   // stop sync if the local issue has been moved from the HELP project
3   if (!firstSync) {
4     syncHelper.unExalateAfterProcessing()
5   }
6   return
7 }
8 // the rest of the script remains untouched and goes after this `if`
```

## Destination side

### Incoming sync

```
1 if (replica.project.key != "HELP") {
2   // stop sync if the remote issue has been moved from the HELP project
3   if (!firstSync) {
4     syncHelper.unExalateAfterProcessing()
5   }
6   return
7 }
8 // the rest of the script remains untouched and goes after this `if`
```

## ON THIS PAGE

Product

-----  
[Stop the sync](#) if the issue has been moved to another project  
[Release History](#)  
[Stop the sync](#) if the issue has been moved to another project  
[Glossary](#)  
[API Reference](#)

[Security](#)  
[Pricing and Licensing](#)

**Resources**

[Academy](#)  
[Blog](#)  
[YouTube Channel](#)  
[Ebooks](#)

**Still need help?**

[Join our Community](#)  
[Visit our Service Desk](#)  
[Find a Partner](#)