

Is it Possible to Keep the Issue Key Same on Both Sides?

Last Modified on 02/03/2026 1:42 pm EST

Keeping the issue key in sync is a bit of a challenge as the destination Jira is allocating the issue in sequence. Jira generates an issue key automatically and expects the key to conform to specific rules.

By default, Jira issue keys (or issue IDs) are of the format **<project key>-<issue number>**, e.g. ABC-123. Exalate is only working through the Jira API. There's no API to manipulate issue keys. This article explains why it is not very simple to have identical issue keys on both sides.

Background

JIRA generates **<issue number>** automatically. Meaning that the first issue in a project will have **<issue number>** set to **1**.

For example: **<project key>** is **ABC** and the first issue in this project will have **ABC-1** as the issue key.

The second issue number will have a number as **<issue number>+1**.

For example, The second issue key will be **ABC-2** (ABC-1+1).

Why It's Not Possible to Use Identical Issue Keys on Both Sides?

You cannot get identical issue keys on both instances because of the following reasons:

Simultaneous Work in Both Instances

Assume that you have two instances with a project ABC that should be kept in sync and a number of issues have already been created.

An issue **ABC-123** on instance A has a twin issue **ABC-123** on instance B.

Then you create a new issue on *instance B*. The new issue on instance B gets **ABC-124** (123+1) as the issue key.

Later someone creates an issue **ABC-124** on instance A. This issue (ABC-124) from instance A needs to be synchronized to instance B. At that moment, the twin issue for the issue **ABC-124** will get the issue key as **ABC-125** (124+1) on instance B, because the issue **ABC-124** already exists in instance B.

An Issue is Removed from an Instance Before the Sync Starts

Assume that you have:

- An instance A with a project ABC. This project has a number of issues that need to be

synced.

- An instance B with a project ABC does not have any new issues created due to synchronization.

You want to sync *project ABC from instance A to project ABC in instance B*.

Someone removed an issue **ABC-2** from *instance A*, and you started synchronization with *instance B, project ABC*.

An issue **ABC-1** from *instance A* is synced into an issue **ABC-1** in *instance B*.

The issue **ABC-2** has been removed from *instance A*, so it's not getting synced.

An issue **ABC-3** from *instance A* is synced into an issue **ABC-2** in *instance B*.

An issue **ABC-4** from *instance A* is synced into an issue **ABC-3** in *instance B*.

The table below shows how the issue key changes in this case.

	Before sync				After sync			
<i>Instance A, Project ABC</i>	ABC-1	ABC-2	ABC-3	ABC-4	ABC-1	ABC-2	ABC-3	ABC-4
<i>Instance B, Project ABC</i>					ABC-1		ABC-2	ABC-3

Workarounds

Depending on the case, you can use one of the options provided below:

- **The global key approach**

In most cases that we encountered until now, the requirement is that the issue can be identified uniquely whenever working with the other team.

Using a 'global key' custom field which is initialized with the parent issue key (the project & the issue on which Exalate has been triggered) is perceived by most as a sufficient workaround.

- **The pre-creation approach**

In cases where you want to migrate issues from one Jira to another, you can pre-create all the issues in the target system (using a script), and then use a [bulk operation to link the source issues with the target issues](#) - ensuring that all the data is transferred

You can copy the original issue key to a custom field in Jira in the twin (or synced) issue. It allows searching for issues by the original issue key, located in the custom field.

Access the twin issue using the remote issue key

- Use the **Remote issue tab** to get information about the twin issue. This tab includes information about synchronized data from the remote issue, including the original issue key.

[All](#) [Comments](#) [Work Log](#) [History](#) [Activity](#) **Remote Issues**

This tab details out what information has been synchronised from the remote issue

Issue from: self using relation: local-blue

Key:	BLUE-95												
Summary:	Status sync check												
Description:	#1												
reporter:	Blue Admin												
Status:	To Do (id=10000)												
Type:	Task (id=10000)												
Priority:	Medium (id=3)												
Project:	<table> <tbody> <tr> <td>Id:</td> <td>10000</td> </tr> <tr> <td>Name:</td> <td>BLUE</td> </tr> <tr> <td>Key:</td> <td>BLUE</td> </tr> <tr> <td>Lead:</td> <td>admin</td> </tr> <tr> <td>Versions:</td> <td>0.2.0 (id=10101) , 1.0.0 (id=10102) , 0.1.0 (id=10100)</td> </tr> <tr> <td>Components:</td> <td>DB (id=10100) , Transport (id=10102) , UI (id=10101)</td> </tr> </tbody> </table>	Id:	10000	Name:	BLUE	Key:	BLUE	Lead:	admin	Versions:	0.2.0 (id=10101) , 1.0.0 (id=10102) , 0.1.0 (id=10100)	Components:	DB (id=10100) , Transport (id=10102) , UI (id=10101)
Id:	10000												
Name:	BLUE												
Key:	BLUE												
Lead:	admin												
Versions:	0.2.0 (id=10101) , 1.0.0 (id=10102) , 0.1.0 (id=10100)												
Components:	DB (id=10100) , Transport (id=10102) , UI (id=10101)												

- Use the [getLocalIssueKeyFromRemoteId](#) syncHelper method in scripts to find a local issue linked to a remote issue.

ON THIS PAGE

[Product](#)

[Background](#)

[Release History](#) ?

[Why It's Not Possible to Use Identical Issue Keys on](#)

[Glossary](#) ?

[Both Sides?](#)

[API Reference](#) ?

[Workarounds](#)

[Security](#) ?

[Pricing and Licensing](#) ?

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](#) ?