

mergeWorkLogs - with custom handling

Last Modified on 01/30/2026 11:20 am EST

Signature

Helper	workLogHelper
Signature	<code>issue.workLogs = workLogHelper.mergeWorkLogs(issue, replica, workLogChangeFn)</code>
Description	Merges the worklogs from two sides and allows adjusting details of every synced workLog. Use a closure function to define how the workLogs should behave.
Introduced in	2.4.0

Example use

- Set the workLog author depending on the previously defined user mapping.

```
//specify workLog author user mapping with the help of a groovy closure
// SETTINGS
def userMapping = [
    "foeey@mycompany.com" : "barry@mycompany.net"
]
def defaultUser = nodeHelper.getUserByEmail("bazy@mycompany.net")
// END: SETTINGS

issue.workLogs = workLogHelper.mergeWorkLogs(issue,replica, { w ->
    w.author = nodeHelper.getUserByEmail(userMapping[w.author?.email]) ?: defaultUser
    w.updateAuthor = nodeHelper.getUserByEmail(userMapping[w.updateAuthor.email]) ?: defaultUser
})
```

- Map the workLog author/updateAuthor by the username. Use it when source and destination Instances use the same Active Directory.

```
Product
//use the local user with the username as the source worklog author/updateAuthor
About Us
issue.workLogs = workLogHelper.mergeWorkLogs(issue,replica, { w ->
Release History
    w.author = nodeHelper.getUserByUsername(w.author?.username)
    w.updateAuthor = nodeHelper.getUserByUsername(w.updateAuthor.username)
Update
API Reference
})
```

[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](#)