

# mergeWorkLogs - with custom handling

Last Modified on 04/15/2024 4:39 am EDT

## Signature

Helper	workLogHelper
Signature	<code>issue.workLogs = workLogHelper.mergeWorkLogs(issue, replica, workLogChangeFn )</code>
Description	<p>Merges the worklogs from two sides and allows adjusting details of every synced workLog.</p> <p>Use a <a href="#">closure</a> function to define how the workLogs should behave.</p>
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
issue.workLogs = workLogHelper.mergeWorkLogs(issue,replica, { w ->
    w.author = nodeHelper.getUserByUsername(w.author?.username)
    w.updateAuthor = nodeHelper.getUserByUsername(w.updateAuthor.username)
})
```

[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](#) 