

Adding basic external script

Last Modified on 01/28/2026 11:32 am EST

This article shows an example of how you can create an example groovy file, add a script, and call it in the Sync Rules.

1. Make sure you have the **scripts** directory. The directory location depends on the issue tracking platform. Custom scripts can only be deployed on Jira Server/Datacenter and nodes which are deployed through the docker deployment approach.

Platform	location
Jira Server	<code><jira-home>/scripts</code>
Jira Datacenter	<code><jira-shared-home>/scripts</code>
Docker based	<code>/opt/<nodename>/data/scripts</code> There could be one of the following values instead of <code><nodename></code> : <ul style="list-style-type: none">◦ <code>snownode</code> for Exalate for ServiceNow.◦ <code>adnode</code> for Exalate for Azure DevOps.◦ <code>hpqcnode</code> for Exalate for HP ALM/QC.
Jira Cloud	Jira Cloud, just as any other cloud node, supports a set of specific scripts. Custom scripts cannot be deployed in this environment. Check out List of external scripts for Jira Cloud for more information.

2. Create **BasicFieldSync.groovy** file with the following code, and store it in the right location on your server. There is no need to restart instance/add-on to enable the external script.

```

class BasicFieldSync
{
    static receive(issue,
        replica,
        nodeHelper,
        commentHelper,
        attachmentHelper) {

        issue.summary    = replica.summary
        issue.description = replica.description
        issue.assignee   = nodeHelper.getUserByUsername(replica.assignee?.username)
        issue.reporter   = nodeHelper.getUserByUsername(replica.reporter?.username)
        issue.labels     = replica.labels
        issue.comments   = commentHelper.mergeComments(issue, replica)
        issue.attachments = attachmentHelper.mergeAttachments(issue, replica)
    }
}

```

3. Call the **BasicFieldSync.groovy** script from the Sync Rules.

- Replace the script in the outgoing sync rules (create and change processors) as below:

Existing script

```

issue.summary    = replica.summary
issue.description = replica.description
issue.assignee   = nodeHelper.getUserByUsername(replica.assignee?.username)
issue.reporter   = nodeHelper.getUserByUsername(replica.reporter?.username)
issue.labels     = replica.labels
issue.comments   = commentHelper.mergeComments(issue, replica)
issue.attachments = attachmentHelper.mergeAttachments(issue, replica)

```

New script

```

BasicFieldSync.receive(
    issue,
    replica,
    nodeHelper,
    commentHelper,
    attachmentHelper
)

```

Product

Now you have one file with basic synchronization rules. You can reuse it in outgoing sync processors `newIssues(create processor)` and for existing issues(`change processor`). If you add new `Code` into the **BasicFieldSync.groovy**, it is automatically executed in your incoming sync rules(`create and change processors`).

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 !\[\]\(cead67df4d82d6c83effe4f8699a7d8f_img.jpg\)](#)