

Working with external scripts in Jira on-premise

JIRA ON-PREMISE

This page describes how you can store advanced groovy scripts outside the [Sync Rules](#) processors.



This feature requires a good understanding of Groovy, the entities of the underlying issue tracker, and Exalate itself.

We offer training and professional services to deliver the solution you're looking for. Please contact sales@exalate.com for more information.

In this section

- [Introduction](#)
- [Where to set up scripts](#)
- [Available external scripts](#)
- [What's next](#)

Introduction

Advanced synchronization scripts usually contain a huge number of code lines. Storing these scripts in the [Sync Rules](#) make it hard to maintain.

With a scripts-specific folder outside of the Sync Rules you can have your own scripts library. You can create your own advanced .groovy scripts and add them into the library. Then call the script from the library in the Sync Rules configuration. Exalate will start to pick external files.

Externalized scripts provide you an ability to configure a complex synchronization case with a reduced amount of code inside the Sync Rules.

It allows making changes to your Exalate configuration easier and version scripts whenever you need it. You can share/copy the Sync Rules between different Connections.

To get autocompletion for all the variables available in the script and the Jira API (such as `com.atlassian.jira.component.ComponentAccessor`), configure the IDE to use shared Maven configuration repository. This helps to improve your scripting experience with Exalate. You can also version your scripts with any major change using version control systems.

Externalized scripts functionality allows to include following files:

- sources
 - groovy
 - compiled .class
 - packaged jar
- resources - plain text and binary resources



The *scripts* directory must have permissions to write and read directories/files.

Where to set up scripts

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.
------------	---

Available external scripts

Exalate documentation includes a lot of example scripts containing approaches on how to implement various use cases. Please note that many of these scripts are used only for example purposes.

Below you can find a list of available external scripts.

Script / Article link	Used to	Supported platforms
Versions.groovy	Sync versions	JIRA SERVER JIRA DATA CENTER
Components.groovy	Sync components	JIRA SERVER JIRA DATA CENTER
ExalateEventListener.groovy DEPRECATED	Subscribe to Exalate Jira issue events: <ul style="list-style-type: none"> • exalated • unexalated • updated • deleted 	JIRA SERVER JIRA DATA CENTER
Status.groovy	Sync statuses	JIRA SERVER JIRA DATA CENTER
IssueLinks.groovy	Sync issue links	JIRA SERVER JIRA DATA CENTER
Move.groovy	Move issues automatically	JIRA SERVER JIRA DATA CENTER
CustomFieldSync.groovy	Sync custom fields	JIRA SERVER JIRA DATA CENTER
SubTask.groovy	Sync sub tasks	JIRA SERVER JIRA DATA CENTER
Epic.groovy	Sync epics on Jira Server / Jira Data Center	JIRA SERVER JIRA DATA CENTER
TempoWorkLogSync	Sync Tempo worklogs	JIRA SERVER JIRA DATA CENTER
SimpleSprintSync.groovy DEPRECATED Check out How to sync sprints in Jira on-premise for relevant info on syncing sprints.	Sync sprints	JIRA SERVER JIRA DATA CENTER

What's next

[Check an example of how you can add an external script](#)

Set up advanced scripting environment

Externalize your existing scripts