

General FAQs

Last Modified on 01/15/2026 2:22 pm EST

How flexible is Exalate?

Exalate uses a Groovy-based scripting engine that helps define the synchronization behavior. You can filter what data to send and how to apply incoming data. It is possible to map fields between instances or transform data from one type of field to another. You can implement the most complex case in Script mode.

Are updates and deletions of comments, attachments, or worklogs supported?

Yes. All standard fields, custom fields, and objects are supported. You can check the issue fields available for synchronization.

How will the solution behave in the case of disaster recovery (where one side needs to restore a backup which is 1-day-old)?

Exalate has an advanced, reliable transactional synchronization engine. Which queues all the changes that need to be applied. And breaks it down into atomic steps that can be retried in case of failure.

What is the storage overhead per issue?

Every replica of an entity is stored on 2 systems. A replica is the result of the transformation of an entity into a *hub issue*. This replica is stored in both instances:

- the source side: Exalate uses replica to detect if relevant changes have been made since the last synchronization
- the destination side: Exalate uses replica to provide the content of the previous version

Is the synchronization status made visible in a simple and straightforward way?

Yes. There is a sync panel in the Jira issue that shows the current status of the synchronization. As for other platforms - you can check the entity status sync from the Exalate admin console.

Does the solution support staging the configuration in another environment without affecting production synchronization?

No. When staging a production environment into a test, it is important that no synchronization happens. Check how you can test Exalate in a staging environment.

What happens when the remote entity is deleted?

When you delete an entity Exalate considers it as a synchronization event. It triggers a number of

events to ensure that future synchronization events are not processed anymore. The synchronization panel indicates that the remote entity has been removed.

Does the third party require administrative access to be able to synchronize?

Not at all - every application administrator can define how 'hub issues' or replicas are translated into the local context.

Are you planning to accommodate other platforms?

Yes. Check the [Exalate integrations](#) to get a full list of compatible platforms.

What type of information can be exchanged?

Any type of information can be exchanged. With the help of the `customKeys` property, you can exchange all the information you want. It allows sending any arbitrary object which will get serialized at one end and deserialized at the other.

What is the licensing and pricing of Exalate?

Synchronization is unlimited between supported platforms as long as Exalate is installed on each instance and has a valid license. Check more details on pricing and licensing.

How can I get a development/community license?

For each valid, paid license you can request a developer's license for a non-production environment. It is the same as the original license including the expiry date.

As for the community license - you can request it in case you have a Jira community license.

Considering JIRA has workflow and customizations for a ticket, how does Exalate handle them during the sync process?

Exalate has a built-in scripting engine to implement almost any use case in a flexible way. Check our use case examples for more details.

Is the remote issue easily accessible?

Optionally. The application administrator can configure how links to the remote issue are represented. In Jira, you can see the remote issue link in the sync panel.

Is it robust and performant enough to scale and support enterprise-level synchronizations?

Exalate is already in use since 2014 at different enterprises. It has been configured as a single-threaded application that processes the synchronization transactions one at a time. This has 2 consequences

1. A minimal system load, as Exalate can be compared to a single user who is working hard to update/retrieve information.
2. It takes time to process transactions (because of the single-threaded nature) and the product is not meant to do large migrations.

One deployment has on average 12000 synchronizations/month without a significant impact on the overall instance performance.

What is the frequency of sync of data/issue details between JIRA & JIRA / JIRA & ALM?

Exalate application is querying the platform for any new change and processes these changes using a single thread. If the number of changes requires more processing time the next cycle is run immediately. But some platforms detect changes by polling in a time interval. Usually, this interval is not bigger than 1 minute. It means that once a change is performed, it can take up to a minute to detect the change and schedule its synchronization.

For the private-public connections, it can take more time as we need to wait for the polling.

Can Exalate for Jira Server work through a proxy?

Yes, it can! You can set up the proxy on your Jira instance. Check Configure an outbound proxy for use in the Jira server for more information.

Do I need to change my Exalate configuration after my Jira or ServiceNow instance is updated?

In general, Jira or ServiceNow updates do not affect your current Exalate configuration.

Minor and Major software releases are run against the latest stable version of Exalate, based on the [basic sync](#) scenario (summary, attachments, comments, description).

[Release History](#)

An [upgrade](#) could affect more advanced integration scripts, so we recommend performing [additional validation](#) in these cases.

[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](#)