

Configure the Connection in Script Mode for ServiceNow

Last Modified on 01/15/2026 6:57 pm EST

Configure the Synchronization (Optional)

Set up Sync Rules

Once you test the Connection, you can choose to configure the synchronization behavior with the help of Sync Rules (Groovy-based scripts) and


1. Edit your Connection.

There are multiple ways to edit your connection:


- Navigate to **Connections** →  **Edit connection.**


Connection	Issues under sync	Last sync	Status
 Company A to Company B Description of the connection lorem ipsum sit amet	123	Issue TEST-777 46 minutes ago	 Active   


- If you just created a connection, select **Configure sync.**

Accept invitation 

Congratulations!

company A




company B


Connection is established.

Configure Sync

2. Set up **Outgoing sync** and **Incoming sync** in the **Rules** tab.

Note: Outgoing sync and Incoming sync scripts:

- are generated by default once the connection has been set up. You can, however, add, edit or remove these scripts to configure your sync according to your requirements.
- are platform specific.
- are present at both the source and the destination instance to give independent control

over what data needs to be sent or received.

In the **Outgoing sync**, you can enter scripts to specify what data you send.

In the **Incoming sync**, you can enter scripts to specify what data you receive.

```
Outgoing sync
1 if (entity.tableName == "incident") {
2   replica.key = entity.key
3   replica.summary = entity.short_description
4   replica.description = entity.description
5   replica.attachments = entity.attachments
6   replica.comments = entity.comments
7   replica.state = entity.state
8 }
9 /*
10  Custom Fields (CF)
11  How to send any field value from the source side to the destination side.
12  1. Add the value to the replica object using the internal field name. Right-click
13  the field in the UI and from the Show option select the field name.
14  2. Uncomment this next statement out and change it accordingly.
15  replica.field_name = entity.field_name
16 */
17
18 /*
19  You can send a field's value using its internal name.
20  For example, to send resolution notes, fetch its internal name:
21  Resolution_Notes -> resolution_notes and add it to the replica as follows:
22  replica.resolution_notes = entity.resolution_notes
23  This also works for other entity types.
24 */
25
26 }
27 // You can sync any other entity using the table name and the entity variable
28 if (entity.tableName == "cndb_ci_business_app") {
29   replica.key = entity.key
30   replica.summary = entity.short_description
31   replica.description = entity.description
32   replica.name = entity.name
33 }
34
35 // Exalate API Reference Documentation: https://docs.exalate.com/docs/exalate-api-reference-documentation
36

Incoming sync
1 if (firstSync) {
2   // For the first sync: Decide the entity you want to create based on the remote issue type
3   entity.tableName = "incident"
4 }
5
6 if (entity.tableName == "incident") {
7   entity.short_description = replica.summary
8   entity.description = replica.description
9   entity.attachments += replica.addedAttachments
10  entity.comments += replica.addedComments
11 }
12 /*
13  Custom Fields (CF)
14  1) Uncomment this next statement out and change it accordingly:
15  entity.field_name = replica.field_name
16 */
17
18 /*
19  Status Synchronization
20  Sync the status according to the mapping [remote incident status: local incident status]
21  Don't include statuses that are not the same in the mapping.
22  def statusMap = [
23    "Open" : "New",
24    "To Do" : "Open"
25  ]
26  def remoteStatusName = replica.status.name
27  entity.state = statusMap[remoteStatusName] ? remoteStatusName
28 */
29 }
30 // Sync any other entity using the table name and the entity variable
31 if (entity.tableName == "cndb_ci_business_app") {
32   entity.short_description = replica.summary
33   entity.description = replica.description
34 }
35
36 // Exalate API Reference Documentation: https://docs.exalate.com/docs/exalate-api-reference-documentation
37
```

Automate your Synchronization through Triggers

Specify **Triggers** if necessary.

With triggers, you can set up automatic sync of entities that fit a specific search query.

You can create **Triggers** by editing the connection or by configuring the sync as shown in the previous step. You can also create them on the Exalate Admin Console under the **Triggers** tab.

Note: For more information, check out [How to create a trigger](#).

Publish the changes made to the Connection

Click **Publish** to save a connection.

Note: You can also use these hotkeys to publish a connection:

- Ctrl+S on Windows or Linux
- Cmd+S on Mac

Note: Changes in existing connections are not be applied before you update the entity. Please make sure to update the entity, if you want to apply changes.

What's Next?

Well done! You have successfully completed your "Getting Started" journey. But there's more.

Options you can consider next:

- Sit back and relax! Sync happens based on the Sync Rules and Triggers you have set up in this step.
- You can also read the [Configuration Guides](#) to see more synchronization examples.
- You can also read the [Exalate API Reference Documentation](#) to learn more about how to work with incoming and outgoing sync scripts (aka Sync Rules).

Product ON THIS PAGE

[About Us](#)

[Configure the Synchronization \(Optional\)](#)

[Glossary](#)

[Publish the changes made to the Connection](#)

[API Reference](#)

[What's Next?](#)

[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](#)