

# Exalate AI Assist: Configure Script Mode Connections [A Step-by-Step Guide]

Last Modified on 01/15/2026 9:31 am EST

## Introduction

Welcome to the Exalate AI Assist! This feature is designed to assist non-technical users in generating and editing sync scripts based on simple prompts. With the AI Assist, you can configure connections without high proficiency in scripting languages.

## Key Features

- **AI-driven script generation:** AI generates a script based on your prompt and the latest sync rules available in the script editor.
- **Interaction in both Sync editors:** Prompts for script generation can be entered in both the Incoming and Outgoing Sync editors.
- **Review and apply changes:** The AI-generated script is displayed in the script editor with highlighted changes. You have the option to insert or discard the changes.

**Note:** the AI Assist feature requires an active internet connection.

## Disabling / Enabling the AI Assist Feature

AI Assist feature in Exalate can be configured via the variable `FEATURE_AI_ASSIST_ENABLED`. By default, this variable is not set.

On Docker, you can disable the AI Assist feature by setting the `FEATURE_AI_ASSIST_ENABLED` variable. To enable it, set `FEATURE_AI_ASSIST_ENABLED=true`.

For the respective issue tracker please check the Environment Variables configuration sections in the [Installation Guides](#) for more details.

For cloud connectors, to enable or disable the AI Assist feature, please [submit a request](#) to the Exalate Support team for assistance.

## How to configure script connection using AI Assist

This guide will walk you through configuring an Exalate script connection using AI Assist.

### Step 1: Access the Sync rule editor

- Open Sync rules editor:
  - Once you have selected or created a connection, click on the "Rules" tab. You will see two editors: the Incoming Sync editor and the Outgoing Sync editor.

- Navigate to the editor where you want to make changes

Note: it's possible to make changes in both editors simultaneously.

✓ **Outgoing sync**

Outgoing sync rules define what information can be sent to the destination instance. Check [the documentation](#) for more details.

```
1 replica.key = issue.key
2 replica.type = issue.type
3 replica.assignee = issue.assignee
4 replica.reporter = issue.reporter
5 replica.summary = issue.summary
6 replica.description = issue.description
7 replica.labels = issue.labels
8 replica.comments = issue.comments
9 replica.resolution = issue.resolution
10 replica.status = issue.status
11 replica.parentId = issue.parentId
12 replica.priority = issue.priority
13 replica.attachments = issue.attachments
14 replica.project = issue.project
15
16 /*
17 | Uncomment these lines if you want to send the full list of versions and components of the source project.
18 | replica.project.versions = []
19 | replica.project.components = []
20 | */
21
22 /*
23 | Custom Fields (CF)
24 | How to send any field value from the source side to the destination side.
25 | 1/ Add the value to the replica object using the Display Name of the specific field.
```

 AI

Specify what you'd like to sync here... For example: I want to sync only internal comments.



The AI Assistant might make mistakes. Please review your scripts after inserting them. To ensure changes take effect, click the publish button.

## Step 2: Use the AI Assist to generate scripts

- Locate the AI Assist input field:
  - In both the Incoming Sync and Outgoing Sync editors, you will find an AI Assist input field with the placeholder text: "Specify what you'd like to sync here...".
- Enter your prompt:
  - Type your prompt in the input field. For example: "I want to sync only internal comments."
- Submit the prompt:
  - After entering your prompt, click the "Submit" button to send it to the AI Assist.

▼


Outgoing sync

Outgoing sync rules define what information can be sent to the destination instance. Check [the documentation](#) for more details.

```
1 replica.key = issue.key
2 replica.type = issue.type
3 replica.assignee = issue.assignee
4 replica.reporter = issue.reporter
5 replica.summary = issue.summary
6 replica.description = issue.description
7 replica.labels = issue.labels
8 replica.comments = issue.comments
9 replica.resolution = issue.resolution
10 replica.status = issue.status
11 replica.parentId = issue.parentId
12 replica.priority = issue.priority
13 replica.attachments = issue.attachments
14 replica.project = issue.project
15
16 /*
17 | Uncomment these lines if you want to send the full list of versions and components of the source project.
18 | replica.project.versions = []
19 | replica.project.components = []
20 */
21
22 /*
23 | Custom Fields (CF)
24 | How to send any field value from the source side to the destination side.
25 | 1/ Add the value to the replica object using the Display Name of the specific field.
```

 AI

Make sure issue type is mapped as follows: Bug = Problem, Issue = Story, Request = Improvement. Any other issue types should be mapped to "Story"



The AI Assistant might make mistakes. Please review your scripts after inserting them. To ensure changes take effect, click the publish button.

### Step 3: Wait for AI processing

- Processing the prompt:
  - Once you submit your prompt, a loading indicator will appear, and the screen will be frozen to show that the AI is processing your request.
- Expected time of processing:
  - The average expected response time depends on the prompt complexity:
    - For short and simple prompts the average time of response is about 40 seconds.
    - For medium and high-complexity prompts the response time can range from 1-5 minutes or slightly longer in some cases.

## Outgoing sync

Outgoing sync rules define what information can be sent to the destination instance. Check [the documentation](#) for more details.

```

1  replica.key           = issue.key
2  replica.type          = issue.type
3  replica.assignee      = issue.assignee
4  replica.reporter      = issue.reporter
5  replica.summary       = issue.summary
6  replica.description   = issue.description
7  replica.labels        = issue.labels
8  replica.comments      = issue.comments
9  replica.resolution    = issue.resolution
10 replica.status        = issue.status
11 replica.parentId      = issue.parentId
12 replica.priority      = issue.priority
13 replica.attachments   = issue.attachments
14 replica.project       = issue.project

15
16 /*
17 | Uncomment these lines if you want to send the full list of versions and components of the source project.
18 | replica.project.versions = []
19 | replica.project.components = []
20 | */
21
22 /*
23 | Custom Fields (CF)
24 | How to send any field value from the source side to the destination side.
25 | 1/ Add the value to the replica object using the Display Name of the specific field.
26 |
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100|

```

Processing your request...

Simple prompts may take a few seconds, while more complex ones could take 3-5 minutes.

Make sure issue type is mapped as follows: Bug = Problem, Issue = Story, Request = Improvement. Any other issue types should be mapped to "Story"

Specify what you'd like to sync here... For example: I want to sync only internal comments.

The AI Assistant might make mistakes. Please review your scripts after inserting them. To ensure changes take effect, click the publish button.

## Step 4: Display AI-generated script

- Generated script display:
  - After processing, the AI-generated script will be displayed in the script editor with highlighted changes.
- AI response message:
  - The AI will notify you when the configuration suggestion is ready.

**Outgoing sync**

Outgoing sync rules define what information can be sent to the destination instance. Check [the documentation](#) for more details.

```

1+// Mapping of issue types from JIRA to ZENDESK
2+ def issueTypeMapping = ['Bug': 'Problem', 'Issue': 'Story', 'Request': 'Improvement']
3+
4+// Assigning values to the replica object
1 5 replica.key           = issue.key
2  -replica.type          = issue.type
6+ replica.type           = issueTypeMapping[issue.type] ?: 'Story'
3 7 replica.assignee      = issue.assignee
4 8 replica.reporter      = issue.reporter
5 9 replica.summary       = issue.summary
6 10 replica.description   = issue.description
7 11 replica.labels        = issue.labels
8 12 replica.comments     = issue.comments
9 13 replica.resolution    = issue.resolution
10 14 replica.status       = issue.status
11 15 replica.parentId     = issue.parentId
12 16 replica.priority      = issue.priority
13 17 replica.attachments  = issue.attachments
14 18 replica.project      = issue.project
15
16 20 /*
17 21     Uncomment these lines if you want to send the full list of versions and components of the source project.
18 22     replica.project.versions = []
19 23     replica.project.components = []
20 24 */
--

```

Discard

Insert Changes

Make sure issue type is mapped as follows: Bug = Problem, Issue = Story, Request = Improvement. Any other issue types should be mapped to "Story"

Your configuration is ready 🎉 If you'd like to make any changes or have more questions, feel free to ask. I'm here to assist you!

**AI**

Specify what you'd like to sync here... For example: I want to sync only internal comments.

The AI Assistant might make mistakes. Please review your scripts after inserting them. To ensure changes take effect, click the publish button.

## Step 5: Review and insert/discard AI suggestions

- Review changes:
  - Carefully review the AI-generated script and the highlighted changes.
- Decision making:
  - Insert Changes: If you are satisfied with the AI suggestions, click the "Insert" button to apply the changes to the script editor.
  - Discard: If you do not want to use the AI suggestions, click the "Discard" button to revert to the latest published sync rules.


**Note:** you cannot edit the script until you insert or discard the AI-suggested changes.

## Step 6: Publish changes


- Finalize script:
  - Once you have reviewed and inserted the changes (if accepted), click the "Publish" button to save the updated script.

- Unfreeze editor:
  - The script editor will be unfrozen, allowing you to make further manual edits if necessary.

#### Product

[About Us](#) 

[Release History](#) 

[Glossary](#) 

[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](#) 

Note: once you publish the changes you cannot revert to the previous version of the script.