The Conversation Mode

Last Modified on 11/16/2026 6:45 am EST

Configure a Connection with Conversation Mode (Aida AI)

Conversation Mode lets you configure a connection by describing the required synchronization behavior in natural language. Aida captures your requirements and continuously generates the underlying Exalate scripts.

Aida is the name used for Exalate's AI-driven capabilities supporting connection setup, configuration, testing, and troubleshooting.

Choose Conversation Mode when you prefer to define the integration through a guided conversation rather than writing Groovy scripts directly.

> **Lab feature**
>
> Conversation Mode is publicly available but remains under active development. Review and test the generated configuration before publishing it.

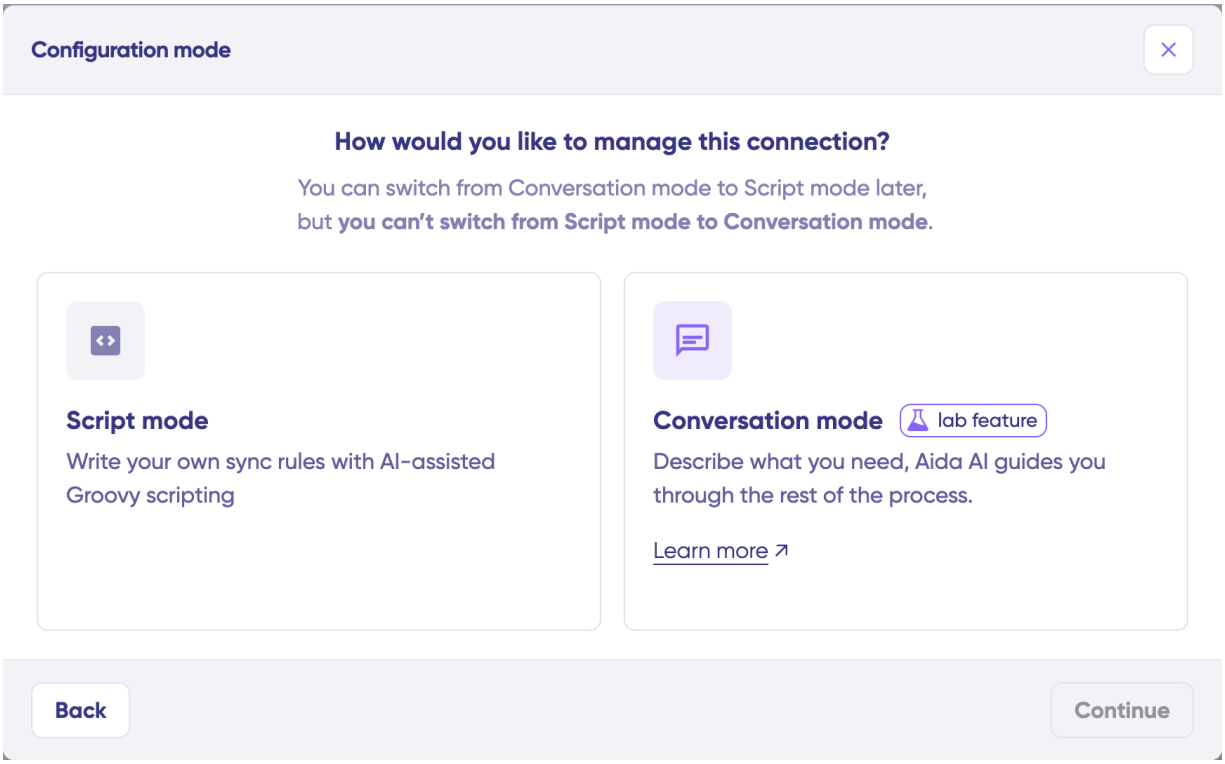
Conversation Mode or Script Mode

Both modes create the same type of Exalate synchronization configuration.

Conversation Mode	Script Mode
Describe the required behavior in natural language	Write and edit Groovy synchronization rules
Aida asks questions and generates the scripts	Aida assists while you edit individual scripts
Best for requirements-driven configuration	Best when direct control over the scripts is required

You can switch a Conversation Mode connection to Script Mode. Currently, this switch is permanent, and the connection cannot be switched back.

This is a temporary limitation. Exalate is developing a hybrid configuration experience in which conversation and direct scripting can be used together.



How Conversation Mode works

The configuration process follows a repeating cycle:

Describe → Clarify → Generate → Test → Refine → Publish

1. Describe how items should be synchronized.
2. Answer Aida's clarification questions. Aida retrieves relevant metadata from the connected ticket systems when needed.
3. Aida captures the requirements, resolves ticket-system references against the available metadata, and generates the scripts.
4. Test the current draft.
5. Correct or extend the requirements through the conversation.
6. Publish the configuration when it is ready.

Conversation is the primary way to manage requirements. The generated Groovy scripts remain the executable configuration underneath.

Configure One Direction At a Time

Each conversation applies to one direction of the connection.

For example, while configuring ServiceNow → Jira, you can describe:

- what ServiceNow sends
- how Jira applies the received data;

- what Jira item should be created or updated.

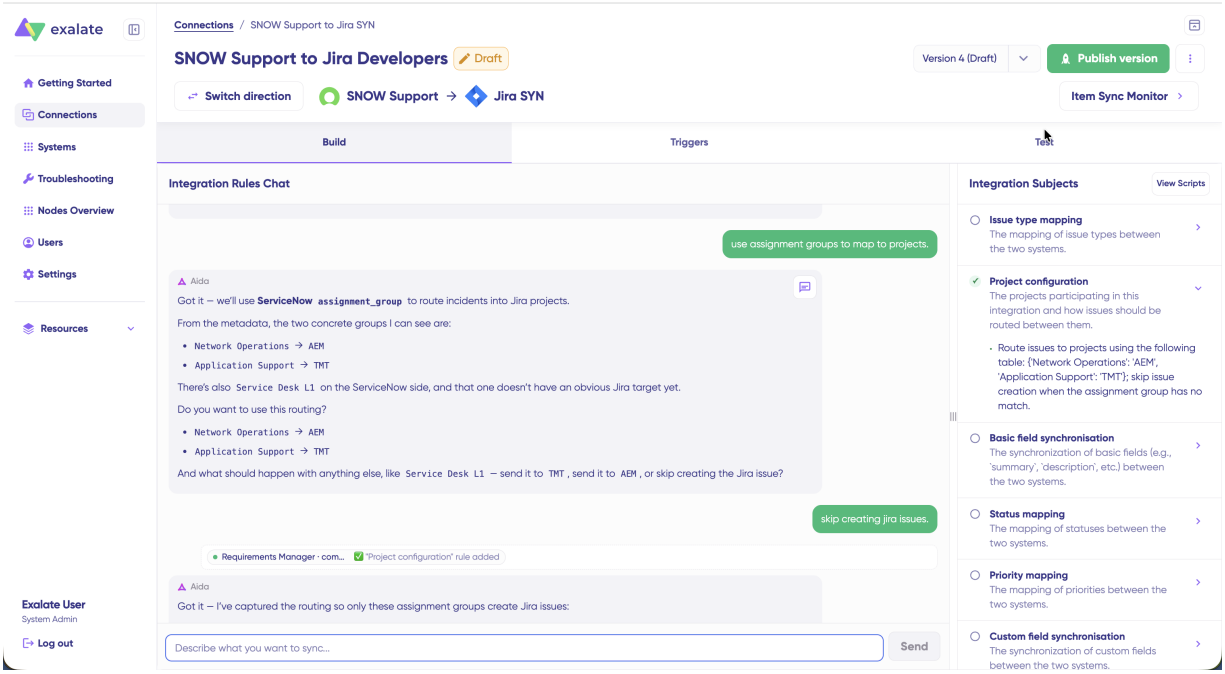
To configure updates in the opposite direction, switch the connection direction and start a separate conversation:

Jira → ServiceNow

When you provide a requirement that belongs to the opposite direction, Aida points this out but does not capture it in the current conversation.

Build the Configuration Through Conversation

There is no required way to begin the conversation. You can start with either a broad goal or a detailed set of requirements.



Start With a General Intent

You can describe the outcome without already knowing how the integration should be configured:

> *"I want our support team in ServiceNow to collaborate with the development team in Jira."*

Aida helps translate this goal into concrete requirements by asking about routing, item types, fields, statuses, comments, and other relevant behavior.

This approach is useful when you understand the business process but have not yet designed the complete synchronization.

Start With Specific Requirements

You can also provide several known requirements at once:

> *"Send escalated ServiceNow incidents to Jira. Route them to different Jira projects based on the ServiceNow assignment group. Synchronize the summary, description, and priority."*

Aida captures the provided information and asks follow-up questions where details are missing. This approach is useful when the integration behaviour has already been designed.

You do not need to describe the complete integration in one message. Start with the information you have and refine it as the conversation progresses. You can correct Aida at any point:

> *"The Application Support assignment group should route to TMT, not AEM."*

Use Ticket-System Metadata in the Conversation

Conversation Mode can retrieve metadata from both ticket systems in a connection. Depending on the connector, this can include projects, objects or tables, item types, fields, statuses, comment visibility settings, and field choice values.

You do not need to look up every exact field name or manually provide lists of available values. You can ask Aida to list the available metadata directly in the conversation. For example:

- “Which projects and issue types are available in Jira?”
- “List the fields for the Support project.”
- “Which values are available for the Salesforce Status field?”

You can also refer to ticket-system configuration in natural language. Aida uses the available metadata to match your description to the corresponding configuration values. If a reference could match more than one value, Aida asks you to clarify which one you mean.

For example, you can ask Aida to propose a status mapping between the two connected systems. Aida compares the status values available on both sides and proposes matches based on their meaning, even when the names differ. Review the proposed mapping before publishing the configuration.

Some metadata requires additional context. For example, fields may depend on a particular Jira project, Salesforce object, or ServiceNow table. If the required scope is missing, Aida asks you to specify it before retrieving the values.

Supported metadata by connector

The metadata available to Aida depends on the connector:

Connector	Metadata available in Conversation Mode
Jira Cloud, including EXJCH	Projects, issue types, statuses, and comment visibility. Fields are retrieved per project. Standard and custom fields appear in one list, and each field is identified as standard or custom.

Connector	Metadata available in Conversation Mode
Salesforce	Objects, statuses, and comment visibility. Fields are retrieved per object. Standard and custom fields are identified, and picklist values are included.
Zendesk	Ticket types, statuses, and comment visibility. Standard and custom fields are included with their choice values, but are not identified separately as standard or custom.
Freshdesk	Ticket types, statuses, and comment visibility. Default and custom fields are included with their choice values, but are not identified separately as default or custom.
Freshservice	Ticket types, statuses, and comment visibility. Default and custom fields are included with their choice values, but are not identified separately as default or custom.
ServiceNow	Tables, statuses, and work notes. Fields are retrieved per table and include the table's own columns and instance-specific customizations. Inherited columns are not included, and fields are not identified separately as standard or custom.
Azure DevOps	Projects and work item types. Fields are retrieved per project and are not identified separately as standard or custom.
Asana	Workspaces, projects, and sections. Custom fields and their choice values are retrieved per workspace. Standard fields are not included.

Metadata availability and limitations

- Aida retrieves metadata as it is needed during the conversation. You can explicitly ask it to list all values in an available metadata category.
- Only metadata available through the connected Exalate node can be retrieved.
- Supported metadata varies by connector, as described above. Aida indicates when requested metadata is unavailable or incomplete.
- Recent changes in a ticket system may not appear immediately because retrieved metadata can be cached.
- Metadata awareness helps Aida identify real configuration values, but proposed requirements, mappings, and generated scripts remain AI-generated. Review and test the configuration before publishing it.

Integration Subjects

Aida organizes captured requirements into integration subjects. These provide an overview of the areas already discussed and those that still require input.

A completed subject means that requirements have been captured for that area. It does not guarantee that the generated configuration is correct or complete.

You can revisit any subject through the conversation and change its requirements.

Background Script Generation

Aida continuously regenerates the underlying scripts when requirements change.

A submitted change is not immediately available for inspection or testing. Exalate must first process it and update the scripts.

While scripts are being generated:

- View Scripts is unavailable;
- Test Run is unavailable.

Wait until generation has completed before inspecting or testing the latest configuration.

View the Generated Scripts

You can inspect the generated outgoing and incoming scripts without leaving Conversation Mode.

This can help you:

- verify how a requirement was implemented;
- troubleshoot an unexpected result;
- review the configuration before publishing;
- determine whether manual scripting is required.

Viewing the scripts does not change the configuration mode. Editing them directly requires switching to Script Mode.

Give Aida Implementation Instructions

Conversation Mode is primarily requirements-driven, but you can also give Aida direct implementation instructions.

For example:

- *"Use an if-else statement instead of a mapping."*
- *"Add a comment using this exact formatting."*
- *"Use this Exalate method to retrieve the value."*

This can help when the intended behaviour is correct, but you want the generated script to use a particular implementation.

Where Aida can apply the instruction, it becomes part of the managed configuration and is retained during later script-generation cycles. Try this before permanently switching to Script Mode.

Configure Triggers

Triggers filter which items you want to synchronize.

Triggers are supported but are configured separately from the conversation.

Conversation Mode defines how those items are processed after synchronization starts.

Trigger configuration works in the same way as in Script Mode.

Test and Refine the Configuration

Use Test Run to validate the current draft against representative source items.

When the result is not correct:

1. Return to the conversation.
2. Explain what should change.
3. Wait for script generation to complete.
4. Run the test again.

Test Run works in the same way as in Script Mode.

Versioning and Publishing

Conversation Mode uses the standard Exalate versioning model.

- Changes are stored in a draft.
- The active configuration remains unchanged until the draft is published.
- Publishing activates the generated scripts.
- Previous versions remain available through the normal version history.

Test the current draft before publishing it.

With publish approval on, submit the draft for review. Approvers see **Differences in the integration rules** and decide. The chat and the rules are frozen while the version is in review.

Switch Permanently to Script Mode

Consider switching to Script Mode when:

- the conversation does not produce the required result;

- the generated scripts require changes that Aida cannot apply;
- the integration requires advanced or unsupported scripting;
- you prefer to manage the configuration directly in Groovy.

Before switching, try describing the required implementation directly to Aida. A script-level instruction may resolve the issue without leaving Conversation Mode.

When you switch, the generated scripts are preserved and become the starting point for the Script Mode configuration.

> Current limitation
>
> The switch to Script Mode is **permanent**. You cannot return this connection to Conversation Mode.

Exalate is working toward a hybrid experience that combines conversation and direct scripting in one configuration mode.

When switching, you can provide the reason why Script Mode was required. This feedback helps Exalate identify scenarios that Conversation Mode does not yet handle successfully.

Provide Feedback

Feedback is especially valuable while Conversation Mode remains a Lab feature.

Annotate an Aida Response

Hover over an Aida response and select the callout action to add an annotation.

Use annotations to report issues such as:

- Aida misunderstood the requirement;
- an important question was not asked;
- the captured requirement is incomplete;
- the generated implementation does not match the requirement;
- the response is unclear.

Every submitted annotation is reviewed by Exalate.

Explain Why You Switched Modes

When switching to Script Mode, provide the reason the conversation could not produce the required result.

This helps identify missing capabilities and recurring failure patterns.

Broader feedback can also be shared through the Exalate Community.

Recommended Workflow

1. Select Conversation Mode.
2. Confirm the connection direction.
3. Describe either the overall goal or the known requirements.
4. Answer Aida's clarification questions.
5. Refer to ticket-system values naturally, or ask Aida to list the available metadata. Clarify the intended project, object, table, field, or value when Aida asks.
6. Wait for script generation to complete.
7. Configure the trigger separately.
8. Run TestRun with representative items.
9. Refine the requirements and test again.
10. Switch direction and configure the reverse flow when required.
11. Publish the tested draft.
12. Switch to Script Mode only when conversation and implementation instructions cannot produce the required result.

Lab Feature Considerations

Conversation Mode is publicly available as a Lab feature.

Aida may:

- misunderstand an ambiguous requirement;
- omit behavior that was not specified;
- generate an incorrect mapping;
- misidentify, omit, or incorrectly map a ticket-system value, particularly when metadata is unavailable, incomplete, or out of date;
- generate scripts that require further correction.

Review the captured requirements, inspect the generated scripts where necessary, and test representative items before publishing.

Product
Overview
Release Notes
Glossary
API Reference
Security
Pricing and Licensing
Resources
Subscribe for a weekly Exalate hack
Academy
Blog
YouTube Channel

ON THIS PAGE

Ebooks

Still need help?

Join our Community

Visit our Service Desk

Find a Partner