Exalate Console

Last Modified on 03/11/2026 5:54 am EDT

New Exalate was formerly called SyncRoom. It is now part of the Exalate Console.

What's New in the Exalate Console?

The Exalate Console provides a unified interface for managing integrations. It delivers consolidated visibility across all connections, safer change handling through Test Run and script versioning, and AI-assisted configuration via Aida, all powered by the same trusted Exalate sync engine.

Overview

The Exalate Console brings integration management to a single place without changing how the underlying sync engine operates. Scripts still run independently per side and per direction. Each system enforces its own rules, logic, and access controls.

What the console adds:

- **Unified visibility** across all nodes and connections from a single interface
- **Test Run** for previewing sync behavior before production deployment
- **Script versioning** with traceable change history and rollback capability
- **Aida** for AI-assisted configuration and troubleshooting
- **Network visualization** showing how nodes are interconnected
- **Bulk operations** for updating multiple connections simultaneously

Console Features

Unified Management Console

Access all your nodes and connections from one place instead of switching between separate consoles for each platform. The console organizes integrations through **Workspaces**, where each workspace groups related connections and systems.

From the console, you can:

- View all connections and their current sync status
- Edit connection settings, scripts, and triggers
- Perform bulk operations across multiple connections
- Import existing connections from Exalate Classic

Network Visualization

A visual map of your integration environment. It shows all nodes (connected systems) and the connections between them, making it straightforward to understand integration architecture, identify dependencies, and spot potential issues.

Side-by-Side View

View script rules, active queues, and errors for both sides of a connection simultaneously. This eliminates the need to switch between systems when diagnosing sync issues or reviewing configurations.

Test Run

Test Run allows you to validate sync scripts before applying them to production data. Select specific items, run the test, and review how the configuration would be applied, including a preview of the replica payload and field mappings.

If the results don't match expectations, adjust the scripts and test again. Deploy only when you're confident the configuration is correct. This prevents errors from reaching live data.

Script Versioning

Every time you publish a configuration change, a new version is created.

You can:

- Work with draft versions without affecting production
- Compare versions to see exactly what changed
- Roll back to a previous version if a change causes problems
- Trace who changed what and when

Versions can be "Active" (currently in production), "Draft" (editable, not yet published), or "Archived" (previous versions no longer in use).

Aida: AI-Assisted Configuration

Aida helps with building sync scripts and troubleshooting errors. It operates within the context of your existing scripts and platform configuration.

For **outgoing scripts**, describe what data should leave your system (e.g., "Exclude attachments" or "Only sync high-priority items"). For **incoming scripts**, describe how incoming data should be applied (e.g., "Map statuses" or "Set a default assignee if the user isn't found").

Aida generates working Groovy scripts with proper field mappings. Review the suggested changes: green highlights indicate additions, red highlights indicate removals, then choose to insert or discard each suggestion.

Aida also explains errors in plain language and suggests context-aware resolutions, reducing troubleshooting time.

> **Note:** As with any AI tool, review generated code before applying it to production.

Import Existing Connections

If you're already running integrations on Exalate Classic, you can import them into the console. Click "Import connection," select your existing Exalate nodes and connections, and continue where you left off. Existing sync rules and configurations carry over. Running integrations are not disrupted during import.

Decoupled Access Control

Integration management access is separated from ticketing system access. The console supports flexible authentication models: API keys, OAuth, Personal Access Tokens, so you can grant someone access to manage integrations without exposing credentials for the underlying systems.

Supported Connectors

Connector availability depends on your plan. The table below shows which connectors are available in the New Exalate Console and Exalate Classic, grouped by plan tier.

Connectors by Plan

Connector	Available On	New Console / Classic Console
Jira Cloud	Starter, Scale, Pro, Enterprise	Both
Jira Service Management (Cloud)	Starter, Scale, Pro, Enterprise	Both
Azure DevOps Cloud	Starter, Scale, Pro, Enterprise	Both
Zendesk	Starter, Scale, Pro, Enterprise	Both
Freshdesk	Starter, Scale, Pro, Enterprise	Both
Freshservice	Starter, Scale, Pro, Enterprise	Both
Asana	Starter, Scale, Pro, Enterprise	Both
GitHub *	Starter, Scale, Pro, Enterprise	Initial setup via Classic
Jira Data Center	Pro, Enterprise	Classic only (Console under consideration)
Azure DevOps Server	Pro, Enterprise	Both
Salesforce	Pro, Enterprise	Both

Connector	Available On	New Console / Classic Console
ServiceNow	Pro, Enterprise	Both
TOPdesk	Pro, Enterprise	Classic (Console coming soon)
Xurrent	Pro, Enterprise	Classic (Console coming soon)
ServiceDesk Plus	Pro, Enterprise	Classic (Console coming soon)
Ivanti	Pro, Enterprise	Classic (Console coming soon)
SolarWinds	Pro, Enterprise	Classic (Console coming soon)
ConnectWise	Pro, Enterprise	Classic (Console coming soon)
HaloITSM	Pro, Enterprise	Classic (Console coming soon)

Custom Connectors

Custom connector development is available as an optional add-on for Enterprise customers, MSPs, and MSSPs. This covers proprietary in-house systems, industry-specific tools, and platforms not listed above.

View All Supported Connectors →

Scripting

The Exalate Console uses the same Groovy-based scripting engine that powers Exalate Classic. Scripts control exactly what data syncs and how it is mapped between systems.

Scripts are divided into:

- **Outgoing script**: Defines what data leaves the source system. Controls which fields, comments, attachments, and custom data are included in the sync payload.
- **Incoming script**: Defines how incoming data is applied to the destination system. Controls field mapping, status transformations, default values, and conditional logic.

Scripts operate independently per side (e.g., Jira side vs. ServiceNow side) and per direction (incoming vs. outgoing). This gives each side full operational control over what data crosses system boundaries.

The **replica** is the data payload exchanged between synced entities. It exists in JSON format and holds the actual values passed between systems. You can inspect the replica during TestRun to verify field mappings before production deployment.

To sync additional values, add the corresponding lines to the appropriate script. To stop syncing specific data (e.g., attachments), remove the relevant script line from the outgoing script.

Security

Exalate implements security controls at multiple levels:

- **ISO 27001:2022 certified** — independently audited information security management
- **Role-based access control (RBAC)** — assign permissions based on user roles to control who can view, edit, or manage integrations
- **Data encryption in transit** — TLS 1.2/1.3 for all data transfers
- **Data encryption at rest** — stored data is encrypted
- **Decoupled authentication** — integration management access is separated from ticketing system credentials. Supports OAuth 2.0, API keys, and Personal Access Tokens
- **Audit trails** — complete logs of configuration changes through script versioning
- **Full script and operational control** — scripts remain separated by direction (incoming/outgoing) and by side. Each system maintains independent control over what data is shared and how it is processed

For complete security documentation, penetration test results, and compliance certificates, visit the Exalate Trust Center.

Pricing

Exalate uses outcome-based pricing. You pay for the number of active items currently in sync — not user seats, not cumulative transactions. Each integration (e.g., Jira ↔ ServiceNow) is billed independently. You can mix plan tiers across integrations based on complexity and connector requirements.

You can find all details here.

Multi-Integration Pricing

Enterprise plans include custom pricing for organizations running more than one integration. This lets you scale your integration network without paying full price for each individual integration. Contact sales for a tailored quote.

View Pricing Details → | ROI Calculator →

Exalate Classic vs. New Exalate Console

Both versions run on the same sync engine. The table below summarizes the key differences in the management experience.

Capability	Exalate Classic	New Exalate Console
Console access	Separate Exalate console per platform	Unified console with side-by-side view
Installation	Add-on/ Exalate installed per system	Web app with ramp-up flow

Capability	Exalate Classic	New Exalate Console
Script editing	AI Assist: AI-assisted scripting	Aida: context-aware script generation and error troubleshooting
Testing	No pre-deployment testing	Test Run: preview sync behavior before production
Versioning	No version history	Full script versioning with rollback and audit trail
Network view	Not available	Network visualization of all nodes and connections
Bulk operations	Supported	Supported
Connectors	All connectors: Jira Cloud & DC, Azure DevOps (Cloud & Server), ServiceNow, Salesforce, Zendesk, Freshdesk, Freshservice, Asana, GitHub, TOPdesk, Xurrent, ServiceDesk Plus, Ivanti, ConnectWise, SolarWinds, HaloITSM	Jira Cloud, Azure DevOps (Cloud & Server), ServiceNow, Salesforce, Zendesk, Freshdesk, Freshservice, Asana, GitHub. Early access connectors (TOPdesk, Xurrent, etc.) coming soon.
Pricing model	User-based (Jira) / Instance-based (others)	Outcome-based — pay per active items in sync per integration

Exalate Classic remains fully supported. The Exalate Console adds unified management on top of the same engine.

Known Limitations

The Exalate Console is being introduced in stages. Current limitations include:

- **Local synchronization** is not yet supported (coming soon)
- **Private connections** (one node behind a firewall) are not yet supported (coming soon)
- **Jira Data Center connector** is under consideration and not currently available
- **User-level permissions** are limited — all users added to the console currently have admin-level access (granular RBAC is planned)

These limitations apply to the current console release and are subject to change as new features are deployed.

FAQs

Can I keep using Exalate Classic? Yes. Exalate Classic continues to operate and is fully supported. You can use both versions simultaneously. Existing integrations on Classic are not

affected by the console.

Do I need to migrate my existing integrations rom Exalate Console? No. You can import existing connections into the console without disrupting running integrations. Import brings over your sync rules and configurations as-is.

Does the console change how my sync scripts work? No. The scripting engine is identical. Scripts written for Exalate Classic work in the console without modification. The console adds Test Run, versioning, and Aida on top of the same scripting layer.

What happens if I publish a bad configuration? Script versioning lets you roll back to any previous version. Test Run lets you validate configurations before publishing, reducing the risk of deploying errors to production.

Which connectors are available in the New Exalate console? Jira Cloud, Jira Service Management, ServiceNow, Azure DevOps Cloud, Azure DevOps Server, Salesforce, Zendesk, Freshservice, Freshdesk, Asana, and GitHub. Additional connectors (TOPdesk, Xurrent, ServiceDesk Plus, Ivanti, ConnectWise, SolarWinds, HaloITSM) are coming soon. Custom connector development is available for Enterprise plans.

How does the outcome-based pricing work? You pay for the number of items actively in sync at any given time per integration. If 50 Jira work items are syncing with ServiceNow, that counts as 50 active items, regardless of how many updates happen to those items. The cost per item decreases as scale increases. Each integration is billed independently, so you can choose different plan tiers per integration. Enterprise plans include custom multi-integration pricing for organizations running multiple integrations.

Is the console secure for cross-company integrations? Yes. Each side of the integration maintains independent control over incoming and outgoing data through separate scripts. Authentication is decoupled from ticketing system credentials. Data is encrypted in transit (TLS 1.2/1.3) and at rest. Exalate is ISO 27001:2022 certified. Visit the Trust Center for full security documentation.

Can I use the console for MSP scenarios? Yes. Connect your instance to multiple client instances. Clients work in their tools. You manage integrations from the unified console with network visualization, bulk operations, and per-connection script customization.

Need Help?

- Exalate Community
- Service Desk
- Product feedback: product@exalate.com

ON THIS PAGE

Exalate Classic vs. New Exalate Console

Known Limitations

FAQs

Need Help?