HaloITSM

Last Modified on 06/29/2026 2:12 pm EDT

CUSTOMER ONBOARDING · HALOITSM · SETUP GUIDE

This guide walks the IT/HaloITSM administrator through registering an OAuth2 API application, scoping it to read-only access on the objects the integration needs, binding it to a dedicated read-only agent, verifying the credentials, and entering them in the Info-Tech portal. It includes the HaloITSM objects that need read-only access so the administrator can complete setup without a separate access-requirements document.

Important — verify against your HaloITSM tenant and documentation
Unlike the other ITSM connectors, a native Fivetran HaloITSM connector has not been confirmed, so this integration is assumed to use a custom REST API / Connector SDK extraction path against Halo's OAuth2-protected REST API. Public Halo documentation confirms the API is token-based OAuth2, JSON, and exposed under each tenant's `/api` resource server — but most object and field details below are **API-derived and need tenant validation**, because the clearest public field schema is a HaloPSA mirror rather than Halo's official ITSM page. Treat menu paths, scope names, and field availability in this guide as a starting point to confirm against your own tenant, not as exact labels. HaloITSM, HaloPSA, and HaloCRM share API surface; this guide covers HaloITSM ticket analytics only.

A Before you start

You will need:

- Administrator access to your HaloITSM instance
- Your HaloITSM API resource server URL
- Your HaloITSM authorization/token endpoint URL
- A dedicated read-only agent to bind the API application to
- Access to the Info-Tech portal where the OAuth client ID, client secret, and tenant URLs will be entered

B HaloITSM objects required

The integration requires read-only API access to the following HaloITSM REST resources (Swagger endpoint names may vary slightly by version):

REQUIRED OBJECTS

- `/Tickets` — ticket records (also called Faults)
- `/Actions` — ticket actions: notes, emails, status changes, and time entries
- `/Users` — end users / requesters
- `/Agent` — agents / technicians
- `/Team` — teams
- `/Client` — clients / customer organizations
- `/Site` — sites

REQUIRED OBJECTS (CONT.)

- `/Status` — ticket statuses
- `/Priority` — priorities
- `/Category` — categories
- `/TicketType` — ticket types
- `/SLA` — SLA definitions
- `/Field` , `/FieldInfo` — custom field definitions

No create, edit, delete, or write-back permissions are required.

Note: satisfaction (CSAT) in HaloITSM is captured at the ticket level via `satisfactionlevel` and `satisfactioncomment` fields on the ticket — no separate survey object is needed. Time entries are read via `GET /Actions?timeentriesonly=true` . Verify the worklog grain, the `timetaken` unit, and CSAT scale per tenant during onboarding.

Important — HaloITSM's permission model
Halo OAuth2 API applications act with the permissions of the agent and the scopes granted to the application. Scope the application to read-only and bind it to a dedicated read-only agent rather than a broad administrator. Grant only the read scope the integration needs.

STEPS IN THIS GUIDE

Step 1 Create a dedicated read-only agent

Configuration → Teams & Agents → Agents → New (exact path varies by version)

Create an agent whose only purpose is API access for this integration, and assign it a read-only role. The OAuth application is bound to this agent, so the application's effective permissions are bounded by the agent's role.

1. **Open Configuration.** Go to the HaloITSM configuration / admin area.
2. **Create or designate a read-only role.** Under **Teams & Agents → Roles** (or the equivalent permissions area), create a role with read/view access to tickets, actions, and the reference objects listed above. Do not grant create, edit, delete, or administration permissions.
3. **Create the integration agent.** Under **Agents**, create an agent named `CIOAnalytics` with email `cioanalytics@yourcompany.com` , and assign the read-only role.

4. **Save the agent.** If your tenant does not allow object-level read scoping on a role, choose the narrowest available read-only role and note this for your Info-Tech onboarding contact.

Step 2 Register an OAuth2 API application

Configuration → Integrations → Halo API / API Applications → New (exact path varies by version)

Halo's REST API is OAuth2-protected. Register an API application to obtain a client ID and client secret the integration uses to request access tokens from your tenant's OAuth2 token endpoint.

1. **Open the Halo API integration area.** Under **Configuration → Integrations**, open the **Halo API** / **API Applications** section.
2. **Create a new API application.** Name it `Info-Tech CIOAnalytics`.
3. **Choose the authentication method.** Select the **Client Credentials** (server-to-server) flow if available, so the integration authenticates with the client ID and secret without an interactive login. If only an agent-bound flow is available, bind it to the `CIOAnalytics` agent from Step 1.
4. **Record the credentials and endpoints.** Save the **Client ID** and **Client Secret**, and note your tenant's **authorization/token endpoint URL** and **API resource server URL** (under `/api`). These go into the Info-Tech portal. Treat the client secret like a password.

Step 3 Scope the application to read-only

On the API application — Permissions / Scopes

Halo API applications are granted scopes that govern what the access token may do. Grant only read scope so the integration cannot modify Halo data.

1. **Grant read scope.** Assign the read-only API scope (for example, a `read:all` or read-tickets scope, depending on what your version exposes). Do not grant write/edit/admin scopes.
2. **Confirm the application is bound to the read-only agent** from Step 1, so both the scope and the agent role constrain access.
3. **Save the application.** If the available scopes are coarser than read-only-per-object, choose the narrowest read scope and rely on the agent role to constrain the rest; note this for your Info-Tech onboarding contact.

Step 4 Verify the OAuth credentials work

Run a test API call from your machine, Postman, or another approved API client.

Before entering the credentials in the Info-Tech portal, confirm the application can mint an access token and read tickets and the reference objects through the Halo REST API. Replace the host and endpoints with your tenant's values.

1. **Mint an access token.** Request a token from your tenant's OAuth2 token endpoint using the client credentials:

   ```
   curl -X POST "https://YOUR_TENANT/auth/token" \
     --header "Content-Type: application/x-www-form-urlencoded" \
     --data-urlencode "grant_type=client_credentials" \
     --data-urlencode "client_id=YOUR_CLIENT_ID" \
     --data-urlencode "client_secret=YOUR_CLIENT_SECRET" \
     --data-urlencode "scope=YOUR_READ_SCOPE"
   ```

   You should receive a JSON response containing an `access_token`. The exact token endpoint path and scope value vary by tenant — use the specific read-only scope configured for your tenant and confirm both in your Halo API configuration. If the request fails, recheck the client ID, secret, scope, and endpoint.

2. **Test a basic ticket read.** Using the access token:

   ```
   curl -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -H "Accept: application/json" \
     "https://YOUR_TENANT/api/Tickets?count=1"
   ```

   A successful response returns ticket JSON. `401 Unauthorized` means the token is invalid or expired; `403 Forbidden` means the application/agent lacks read access.

3. **Test the time-entries (worklog) read.** Confirm the filtered Actions endpoint returns time entries:

   ```
   curl -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
     -H "Accept: application/json" \
     "https://YOUR_TENANT/api/Actions?timeentriesonly=true&count=1"
   ```

   Confirm the response includes timing fields such as `timetaken`. If the endpoint or filter is not recognized, flag it with your Info-Tech onboarding contact.

4. **Test reference reads.** Confirm each returns data: `/api/Users`, `/api/Agent`, `/api/Team`, `/api/Client`, `/api/Site`, `/api/Status`, `/api/Priority`, `/api/Category`, `/api/TicketType`, `/api/SLA`. If any return `403 Forbidden`, add the minimum read scope/role permission for that object and test again.

Step 5 Enter the credentials in the Info-Tech portal

Info-Tech portal

> The OAuth client secret gives access to your HaloITSM data. Treat it like a password. **DO NOT SEND THIS TO INFO-TECH.** See the **Submit to Info-Tech Portal** article for the portal submission steps. You will also need your HaloITSM API resource server URL and OAuth2 token endpoint URL.

NOTES ON THIS DOCUMENT

- HaloITSM navigation and available scope names vary by version. If a menu item is not in the exact location shown, use the closest matching Configuration, Integrations, or Agents page.
- Because a native Fivetran HaloITSM connector has not been confirmed, treat object names, field names, and endpoint paths as starting points to validate against your tenant's schema during onboarding.
- Questions about anything in this guide can be directed to your Info-Tech onboarding contact.