ServiceNow

Last Modified on 06/25/2026 2:09 pm EDT

CUSTOMER ONBOARDING · SERVICENOW · SETUP GUIDE

This guide walks a customer IT or ServiceNow administrator through creating a dedicated read-only integration account, assigning read-only access to the required ServiceNow tables, registering an OAuth 2.0 Client Credentials application, verifying Table API access, and entering the connection details through the Info-Tech submission flow. It includes the ServiceNow tables and fields needed so administrators can complete setup from a single guide.

Reviewed against current public documentation
This version has been aligned to the current ServiceNow Client Credentials workflow, Table API behavior, non-interactive-user guidance, and the current Fivetran ServiceNow connector requirements available as of June 10, 2026.

A Before you start

You will need:

- Administrator or delegated administrator access to your ServiceNow instance
- Permission to manage OAuth inbound integrations and service-account users
- Permission to grant read access to the required ServiceNow tables
- A dedicated service-account email address you control (e.g., `cioanalytics@yourcompany.com`)
- Access to the Info-Tech portal where the instance URL and OAuth credentials will be entered

B ServiceNow tables required

The integration requires read-only access to the following ServiceNow tables. Required tables are needed for any ServiceNow tenant; conditional tables are only needed if your instance uses the corresponding ITSM module.

REQUIRED TABLES

- `task` — parent table (shared ticket fields)
- `incident` — incident records
- `sys_user` — user / agent reference
- `sys_user_group` — assignment group reference
- `sys_user_grmember` — group membership bridge
- `sys_user_has_role` — user-to-role bridge (agent role names)
- `cmn_department` — department reference
- `cmn_location` — location reference
- `contract_sla` — SLA definitions
- `task_sla` — SLA instances per task
- `task_time_worked` — time-worked / worklog entries per task
- `sys_journal_field` — work notes / comments thread

CONDITIONAL TABLES

- `change_request` — if Change Management is enabled
- `problem` — if Problem Management is enabled
- `sc_request` — if Service Catalog is enabled
- `sc_req_item` — with sc_request
- `sc_task` — with sc_request
- `asmt_assessment_instance` — if Assessment & Survey (CSAT) is enabled
- `asmt_metric_result` — with asmt_assessment_instance
- `asmt_metric` — with asmt_assessment_instance
- `asmt_metric_type` — with asmt_assessment_instance

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

Important — ServiceNow uses Class Table Inheritance
ServiceNow stores ticket data across a parent table (`task`) and several child tables (`incident` , `change_request` , `problem` , etc.). Shared fields like `sys_id` , `number` , `state` , and `opened_at` live on the parent. Subclass-specific fields like `caller_id` , `contact_type` , `category` , and `close_code` live on the child tables only. Both the parent and the relevant child tables must be granted read access for the integration to work correctly — granting access only to `task` would result in NULL values for many fields.

STEPS IN THIS GUIDE

Step 1 Create a dedicated read-only role

All > System Security > Users and Groups > Roles > New

Create a dedicated role rather than reusing a broad built-in role. This keeps the integration's permissions scoped, auditable, and easy to revoke without affecting other ServiceNow users.

1. **Open the role list.** In the navigator, type `roles.list` and press Enter, or browse to **System Security → Users and Groups → Roles**.
2. **Click New.** Create a new role.
3. **Name the role clearly.** Suggested name: `x_cioanalytics_read_only` or `infotech_cioanalytics_read_only` . Follow your organization's naming convention for custom roles. Description: *Read-only access for the Info-Tech Customer Data Store integration.*

4. **Save the role.** Do not add broad inherited roles unless your security team has approved them. Table and field access will be handled in Step 2.

Step 2 **Grant read-only table access to the role**

All > System Security > Access Control (ACL)

ServiceNow Access Control Lists govern record and field access. Grant only the read access needed by the integration. Do not relax existing ACL conditions or grant create, write, or delete access.

1. **Open the ACL list.** In the navigator, type `sys_security_acl.list` and press Enter.
2. **For each required table**, create or update a record ACL with these values, following your organization's ACL standards:
   - **Type:** record
   - **Operation:** read
   - **Name:** the table name, such as `task`, `incident`, or `sys_user`
   - **Roles:** add the dedicated read-only role created in Step 1
3. **Repeat for all required tables:** `task`, `incident`, `sys_user`, `sys_user_group`, `sys_user_grmember`, `sys_user_has_role`, `cmn_department`, `cmn_location`, `contract_sla`, `task_sla`, `task_time_worked`, `sys_journal_field`, and `sys_audit_delete`.
4. **Also grant read access to the schema metadata tables.** The connector reads these during schema discovery to learn which tables and fields exist. Without them the connection fails before any data is read, typically with a message such as "User does not have access to sys_db_object and/or sys_dictionary tables." For each metadata table — `sys_db_object` (the table dictionary), `sys_dictionary` (the field dictionary), and `sys_glide_object` (the field-type dictionary) — create two read ACLs:
   - A **record ACL** on the table itself — Type `record`, Operation `read`, Name `sys_db_object` (and again for `sys_dictionary` and `sys_glide_object`).
   - A **field ACL** covering all fields — Type `record`, Operation `read`, Name `sys_db_object.*` (and again for `sys_dictionary.*` and `sys_glide_object.*`). On hardened instances the table-level ACL alone can still leave fields denied, so the wildcard field ACL is required.

   Add the dedicated read-only role from Step 1 to each of these ACLs. The role is assigned to the integration user in Step 3, and the OAuth application is bound to that same user in Step 4, so no separate role assignment is needed here.
5. **Add conditional tables if used:** if your instance uses Change Management, also add `change_request`; if Problem Management, add `problem`; if Service Catalog, add `sc_request`, `sc_req_item`, and `sc_task`; if the Assessment & Survey (CSAT) module is in use, add `asmt_assessment_instance`, `asmt_metric_result`, `asmt_metric`, and `asmt_metric_type`.
6. **Check field-level ACLs.** Some hardened instances restrict fields such as `sys_user.email`, phone fields, or journal fields. If field-level ACLs block required fields, grant read access only to the specific fields needed by this guide.
7. **Do not grant write, create, or delete operations.** The integration is read-only.

8. **Alternative:** assigning a broad role such as `itil` may be simpler but grants more access than required. Use this only if your security team approves the broader permission model.

Step 3 Create a dedicated non-interactive service-account user

All > User Administration > Users > New

Create a service account whose only purpose is API access for this integration. A dedicated non-interactive account avoids tying the connection to a human user and supports clean deactivation if access must be revoked.

1. **Open the user list.** In the navigator, type `sys_user.list` and press Enter.
2. **Click New.**
3. **Fill in user details.** Suggested values:
   - **User ID:** `cioanalytics`
   - **First name:** `CIOAnalytics`
   - **Last name:** `Integration`
   - **Email:** `cioanalytics@yourcompany.com`
   - **Active:** checked
   - **Identity type:** `Machine` — marks the account as a non-interactive machine identity rather than a person.
   - **Time zone:** `GMT`
4. **Save the user.**
5. **Assign the read-only role.** Open the user record, scroll to the **Roles** related list, click **Edit**, and add the dedicated read-only role from Step 1. Also add the `personalize_dictionary` role.
6. **Set and store a strong password if your instance requires one.** The OAuth Client Credentials token request uses the OAuth client ID and client secret. However, some downstream connector configurations may also require the service-account username and password. Store the password securely and enter it only through the approved Info-Tech submission flow if the final portal requires it.

Step 4 Register an OAuth 2.0 Client Credentials application

All > System OAuth > Application Registry > New

Use OAuth 2.0 Client Credentials Grant for this trusted, server-side integration. This flow lets the integration request an access token using a client ID and client secret, without an interactive user login. Newer instances may expose the same setup through **Machine Identity Console → Inbound integrations**.

1. **Choose the setup path available in your instance.** Many instances use **System OAuth → Application Registry**. Newer instances may expose the same inbound OAuth setup through **Machine Identity Console → Inbound integrations**.
2. **Create the OAuth client.** If using Application Registry, click **New** and select **Create an OAuth API endpoint for external clients**.

3. **Name the application.** Suggested name: `Info-Tech CIOAnalytics`.
4. **Set the grant type.** Set **Default Grant Type** or **Grant Type** to `Client Credentials`. Do not enable Authorization Code or OIDC-specific options for this integration.
5. **Enable the inbound Client Credentials grant for the instance.** ServiceNow gates the inbound Client Credentials flow behind a system property that is off by default. If it is not enabled, the token request in Step 5 will fail even when the OAuth client is configured correctly. In the navigator, type `sys_properties.list` and press Enter, then search for `glide.oauth.inbound.client.credential.grant_type.enabled`. If the property exists, set its **Value** to `true`. If it does not exist, click **New** and create it with Name `glide.oauth.inbound.client.credential.grant_type.enabled`, Type `true | false`, and Value `true`. This typically requires the admin role.
6. **Bind the OAuth client to the service account.** Set **OAuth Application User** or **User** to the `cioanalytics` service account created in Step 3. If the field is not visible on the OAuth client form, use **Configure → Form Layout** and add **OAuth Application User** and **Default Grant Type**.
7. **Configure the Auth scope.** Scroll to the **Auth scope** section of the application form. Under **Configure auth scopes**, set the **Auth scope** dropdown to `useraccount`. Leave the **Limit authorization to the following APIs** field blank. Under **Scope validation settings**, the **Allow access only to APIs in selected scope** checkbox will be automatically disabled — this is expected behaviour when the `useraccount` scope is selected.
8. **Leave Redirect URL blank unless your instance requires a value.** Redirect URLs are for browser-based authorization-code flows. This guide uses Client Credentials Grant.
9. **Do not configure refresh tokens for this guide.** Client Credentials integrations request new access tokens using the client ID and secret. Refresh-token handling should only be added if your security architecture explicitly requires it.
10. **Save the application.** Record the **Client ID**, securely reveal and record the **Client Secret**, and keep your **ServiceNow instance URL**, such as `https://yourinstance.service-now.com`, for Step 5 and Step 6.

Step 5 **Verify the OAuth credentials and Table API reads**

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

Before entering credentials in the Info-Tech submission flow, confirm that the OAuth client can mint an access token and that the service account can read the required ServiceNow tables through the Table API.

1. **Mint an access token.** Run from a terminal, replacing the placeholders:

```
curl -X POST "https://yourinstance.service-now.com/oauth_token.do" \
  --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"
```

You should receive a JSON response containing an `access_token`, `token_type`, and `expires_in`. If you get `invalid_client`, confirm the Client ID and Client Secret. If you get `unsupported_grant_type` or `access_denied`, confirm the OAuth client is configured for Client Credentials and is bound to the service account. If the response says the property `glide.oauth.inbound.client.credential.grant_type.enabled` must be defined and set to true, the instance-level Client Credentials grant has not been enabled — set that system property to true as described in Step 4, then retry.

2. **Test a basic table read.** Using the access token from the previous step:

```
curl -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  -H "Accept: application/json" \
  "https://yourinstance.service-now.com/api/now/table/incident?sysparm_limit=1"
```

   A successful response returns a JSON `result` array. `401 Unauthorized` usually means the token is invalid or expired; `403 Forbidden` usually means the service account lacks read access to the requested table or field.

3. **Test required table reads.** Confirm each endpoint returns a successful response. An empty result can be acceptable for tables with no records, but authentication and authorization should still succeed:
   - `/api/now/table/task?sysparm_limit=1`
   - `/api/now/table/incident?sysparm_limit=1`
   - `/api/now/table/sys_user?sysparm_limit=1`
   - `/api/now/table/sys_user_group?sysparm_limit=1`
   - `/api/now/table/sys_user_grmember?sysparm_limit=1`
   - `/api/now/table/sys_user_has_role?sysparm_limit=1`
   - `/api/now/table/cmn_department?sysparm_limit=1`
   - `/api/now/table/cmn_location?sysparm_limit=1`
   - `/api/now/table/contract_sla?sysparm_limit=1`
   - `/api/now/table/task_sla?sysparm_limit=1`
   - `/api/now/table/task_time_worked?sysparm_limit=1`
   - `/api/now/table/sys_journal_field?sysparm_limit=1`
4. **Test the schema metadata tables.** These must succeed for schema discovery; a 403 here is the cause of the “User does not have access to sys_db_object and/or sys_dictionary tables” error. Confirm each returns a successful response:
   - `/api/now/table/sys_db_object?sysparm_limit=1`
   - `/api/now/table/sys_dictionary?sysparm_limit=1`
   - `/api/now/table/sys_glide_object?sysparm_limit=1`
5. **Test conditional tables if used.** If your instance uses Change Management, Problem Management, Service Catalog, or the Assessment & Survey (CSAT) module, also confirm:
   - `/api/now/table/change_request?sysparm_limit=1`
   - `/api/now/table/problem?sysparm_limit=1`
   - `/api/now/table/sc_request?sysparm_limit=1`
   - `/api/now/table/sc_req_item?sysparm_limit=1`
   - `/api/now/table/sc_task?sysparm_limit=1`
   - `/api/now/table/asmt_assessment_instance?sysparm_limit=1`

- `/api/now/table/asmt_metric_result?sysparm_limit=1`
- `/api/now/table/asmt_metric?sysparm_limit=1`
- `/api/now/table/asmt_metric_type?sysparm_limit=1`

6. **Test display and raw values.** Run:

```
curl -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  -H "Accept: application/json" \
  "https://yourinstance.service-now.com/api/now/table/contract_sla?sysparm_limit=1&sysparm_display_value=all"
```

The response should include both raw and display values. This is useful when validating reference fields, choice fields, dates, and SLA duration fields.

Step 6 **Enter connection details through the Info-Tech submission flow**

Info-Tech submission flow — interim guidance until the formal portal is available

> The OAuth client secret and any service-account password are sensitive credentials. Do not send them to Info-Tech by email, chat, ticket attachment, or screen share. Use only the approved Info-Tech submission flow. See the **Submit to Info-Tech Portal** article for interim placeholder guidance until the formal site is available.

NOTES ON THIS DOCUMENT

- ServiceNow navigation varies by release, role, and enabled experiences. This guide includes both the legacy **Application Registry** path and the newer **Machine Identity Console** path where relevant.
- If your security team requires more granular access, prefer field-level read ACLs and a dedicated custom role over broad built-in roles.
- Questions about business purpose, data scope, or onboarding sequencing can be directed to your Info-Tech onboarding contact.