

ServiceNow Flow Connector Installation & Setup Guide

Install, connect, and call your first ServiceNow operation from Flow

NexGen Architects | July 2026

How to install the ServiceNow Flow Connector managed package into a Salesforce org, connect it to a ServiceNow instance, and call your first ServiceNow operation from Flow Builder — with no middleware and no code.

KEY FACTS

Product	ServiceNow Flow Connector — a Salesforce managed package that adds ServiceNow actions to Flow Builder.
Package version	1.0.0 (connector model declares <code>since: R258</code>).
Install from	AppExchange listing.
Operations	45 ServiceNow operations across Incidents, Change Requests, Service Requests, Requested Items, Tasks, CMDB, Attachments, Journal Entries, Knowledge, and Users.
Authentication	4 connection types — Basic Auth, plus OAuth 2.0 Authorization Code, Client Credentials, and Password grants.
Infrastructure	None. Calls go directly from Salesforce to your ServiceNow instance. NexGen Architects operates no servers in the data path.
Setup time	Approximately 20–30 minutes for a first-time OAuth setup.

Before you begin: what you need

Setup touches two systems. Gather all six items below before you start — the most common cause of a failed first install is discovering mid-way that you lack the ServiceNow admin rights needed to register an OAuth application.

In Salesforce

- An org on a platform release that satisfies the connector's declared minimum (`since: R258`). Confirm your org's release train before installing.
- An edition that includes Flow Builder and Named Credentials — Enterprise, Unlimited, Performance, or Developer Edition.
- The **Customize Application** and **Modify All Data** permissions, or the System Administrator profile, to install a managed package and create Named Credentials.

In ServiceNow

- Your instance subdomain — the `acme` in `https://acme.service-now.com`.
- An integration user account whose roles grant access to the tables you intend to use. Roles are the real access-control boundary.
- For OAuth: the `admin` or `oauth_admin` role, needed to create an entry in the Application Registry.

Step 1 — Install the managed package

1. **Open the AppExchange listing.** Go to the ServiceNow Flow Connector listing and choose **Get It Now**.
2. **Install into a sandbox first.** Select **Install in Sandbox**. Validate the full setup there before touching production — this is standard practice for any managed package, and it lets you test ServiceNow credentials without risking live records.
3. **Choose who gets access.** **Install for Admins Only** is the recommended starting point. You will grant broader access deliberately in Step 4 by assigning a permission set, rather than opening the package to all users up front.
4. **Approve the remote site access.** The package calls out to your ServiceNow instance. Approve third-party access when prompted; you will bind the actual hostname in Step 2 via a Named Credential.
5. **Confirm the install.** In Setup, open **Installed Packages** and verify the connector appears at version 1.0.0.

Step 2 — Create the External Credential and Named Credential

The connector never stores a username, password, or token itself. It authenticates through a Salesforce Named Credential, which points at an External Credential holding the secrets. This is what keeps ServiceNow credentials inside the Salesforce encrypted store and out of your flows.

This guide gives the short path. For full, screen-by-screen instructions covering all four authentication methods — including the ServiceNow-side OAuth Application Registry setup — see the companion document, **Configuring Salesforce Named Credentials for ServiceNow (OAuth 2.0 & Basic)**.

1. **Create an External Credential.** In Setup, go to **Security** → **Named Credentials** → **External Credentials** → **New**. Choose an authentication protocol: OAuth 2.0 (recommended) or Basic Authentication.
2. **Add a Principal.** Under the External Credential, create a Principal — this is the identity that holds the ServiceNow secret. Name it something durable, such as `ServiceNow_Integration_User`.
3. **Create the Named Credential.** Go to the Named Credentials tab, choose **New**, and set the URL to your instance root: `https://<your-instance>.service-now.com`. Link it to the External Credential you just created.
4. **Enable the callout settings.** Tick **Generate Authorization Header**, and enable **Allow Formulas in HTTP Header** and **Allow Formulas in HTTP Body** if your flows will build dynamic requests.

Step 3 — Authorize the connection

For the OAuth 2.0 Authorization Code flow, the Principal must be authorized once, interactively, before any flow can use it.

1. Open the External Credential and find your Principal in the list.
2. Choose **Authenticate** from the row-level menu. Salesforce redirects you to your ServiceNow instance at `https://<instance>.service-now.com/oauth_auth.do`.
3. Sign in as the integration user — **not as yourself**. Whoever signs in here is the identity every flow will act as, and every record the connector creates will be attributed to that user.
4. Confirm the status shows **Authenticated**. Salesforce exchanges the authorization code for an access token at `/oauth_token.do` and stores it. Refresh is handled automatically from then on.

Step 4 — Grant users access to the credential

A user cannot run a flow that uses a Named Credential unless they have been granted access to its Principal. This is a genuine access-control decision, not a formality — it is how you control who can reach ServiceNow through Salesforce.

1. Create or open a permission set in Setup, for example `ServiceNow Integration Users`.
2. **Add the External Credential Principal**. In the permission set, open **External Credential Principal Access** and enable your Principal.
3. Assign the permission set to the users — or, more commonly, to the Automated Process / running user — that will execute the flows.

Step 5 — Verify the connection

The connector ships with a built-in connection test. It issues a `GET` against `/api/now/table/sys_user` and treats HTTP 200 as success, reporting `Connection test succeeded`.

You can reproduce the same check by hand to confirm credentials independently of Salesforce. Run this from a terminal, substituting your instance and integration user:

```
curl -u '<integration-user>:<password>' \
  'https://<instance>.service-now.com/api/now/table/sys_user?sysparm_limit=1'
```

A JSON body containing a `result` array confirms the account can authenticate and read. An HTML login page in the response body means the request was not authenticated at all — see the troubleshooting table below.

Step 6 — Use the connector in Flow Builder

1. Create a flow — a Record-Triggered Flow on Case is the most common starting point.
2. Add an **Action** element and search the action palette. In version 1.0.0 the connector's actions are listed under the category `servicenow`, which is the `displayName` declared in the shipped connector model.
3. Pick an operation — for example `createIncident`, which maps to `POST /api/now/table/incident`. The full list is in the companion **Supported ServiceNow Operations Matrix**.
4. Select your connection, then map the inputs. For `createIncident`, the operation takes a single `incident` body input carrying the fields to set — `short_description`, `description`, `urgency`,

`caller_id` , and so on.

5. Debug the flow and confirm the record appears in ServiceNow. Store the returned `sys_id` on your Salesforce record so later flows can update the same ServiceNow record.

Troubleshooting

Symptoms, causes, and fixes for the most common setup failures.

Symptom	Most likely cause	Fix
HTTP 401 Unauthorized	Credentials are wrong, or the OAuth Principal was never authorized / its refresh token was revoked.	Re-run Authenticate on the External Credential Principal. For Basic Auth, confirm the password has not expired in ServiceNow.
HTTP 403 Forbidden	The integration user authenticated successfully but lacks the ServiceNow role for that table.	Grant the role — <code>itil</code> for incidents and tasks, appropriate CMDB roles for configuration items. Fix in ServiceNow, not Salesforce.
HTTP 200 but the result array is empty	ACLs are silently filtering the rows. This is ServiceNow's default behaviour for denied reads and is not an error.	Log in to ServiceNow as the integration user and run the same query in the UI. If you see no rows there, the problem is roles — not the connector.
HTTP 404 Not Found	Wrong <code>sys_id</code> , a table that does not exist on the instance, or a path accidentally appended to the Named Credential URL.	Confirm the Named Credential URL is the bare instance origin with no path. Verify the <code>sys_id</code> exists.
HTTP 400 Bad Request	A malformed <code>sysparm_query</code> , or a field value rejected by ServiceNow — for example text sent to a numeric <code>urgency</code> field.	Test the query string in the ServiceNow REST API Explorer first, then copy the working form into the flow.
An HTML login page comes back instead of JSON	The request reached ServiceNow unauthenticated, or SSO intercepted it. Common when the instance enforces SSO for all logins.	Ensure the integration user is exempt from SSO-only enforcement, or use OAuth. ServiceNow admins typically set <code>glide.authenticate.sso.redirect.idp</code> exemptions for integration accounts.
Works in Debug, fails when the flow runs for real	The running user (often Automated Process) lacks External Credential Principal Access.	Assign the permission set from Step 4 to the flow's running user.
Callout not allowed from this context	The flow attempts a callout in the same transaction as a DML operation, before the commit.	Move the ServiceNow action into an asynchronous path in the record-triggered flow. This is a Salesforce platform rule, not a connector limitation.

Frequently asked questions

Does the ServiceNow Flow Connector require MuleSoft, middleware, or any server to run?

No. The ServiceNow Flow Connector requires no middleware, no MuleSoft runtime, and no infrastructure of any kind. It is a Salesforce managed package that runs inside your Salesforce org and calls the ServiceNow REST API directly over HTTPS via a Named Credential. NexGen Architects operates no servers in the data path and never receives your data.

How long does it take to install and configure the ServiceNow Flow Connector?

A first-time setup using OAuth 2.0 typically takes 20–30 minutes. Installing the managed package takes about 5 minutes. Most of the remaining time is spent on the ServiceNow side, registering an OAuth application in the Application Registry and confirming the integration user's roles. Subsequent installs into other orgs are considerably faster once the ServiceNow OAuth application already exists.

Which Salesforce editions support the ServiceNow Flow Connector?

Any Salesforce edition that includes Flow Builder and Named Credentials — Enterprise, Unlimited, Performance, and Developer Edition. Editions without Named Credential support cannot authenticate the connector. The connector model also declares a minimum platform release of `R258`; verify your org's release train before installing.

Do I need to write any Apex code to use the ServiceNow Flow Connector?

No. The connector is designed to be used entirely from Flow Builder with no code. All 45 operations are exposed as declarative Flow actions with typed inputs and outputs. You configure them by mapping fields in the Flow Builder UI. Apex is not required for any supported operation.

Where are my ServiceNow credentials stored when using the connector?

Your ServiceNow credentials are stored in Salesforce's encrypted Named Credential store and are never held by the connector or by NexGen Architects. The connector requests a callout through the Named Credential; Salesforce injects the authorization header at call time. Credentials are not visible in flow definitions, not written to debug logs, and not transmitted to any third party.

My flow returns no ServiceNow records but reports no error. What is wrong?

This is almost always ServiceNow ACLs silently filtering the results, not a connector fault. When the integration user lacks read access to a table, ServiceNow commonly returns HTTP 200 with an empty `result` array rather than an explicit 403. To confirm: log in to ServiceNow as the integration user and run the same query in the UI. If no rows appear there, grant the user the required role — `itil` for incidents, for example — and re-run the flow.

Can I install the ServiceNow Flow Connector in a sandbox before production?

Yes, and it is strongly recommended. Choose **Install in Sandbox** from the AppExchange listing and complete the full setup — including the ServiceNow OAuth registration — against a ServiceNow sub-production or developer instance. Because the connector holds no state of its own, promoting to production means repeating the Named Credential setup against your production instance; there is no data migration step.

What happens to my flows if I uninstall the ServiceNow Flow Connector?

Uninstalling the managed package removes the connector's Flow actions, so any flow still referencing them will fail to save or activate. Remove or replace the ServiceNow actions in your flows before uninstalling. Because the connector stores no data, uninstalling destroys no records in either Salesforce or ServiceNow — records already created in ServiceNow remain untouched.