servicenow

ServiceNow CLI

User Guide

servicenow

Table of Contents

servicenow

What is the ServiceNow CLI?

The ServiceNow Command Line Interface (CLI) is an extensible framework that enables users to communicate with an instance from any client machine using the command line.

Installing the CLI

The latest version of the ServiceNow CLI is **1.0.0**. To find your installed version and see if you need to update, run `snc version`.

MacOS

Install the ServiceNow CLI on MacOS.

Prerequisites

- The ServiceNow CLI supports Apple-supported versions of 64-bit macOS.

Install the ServiceNow CLI on MacOS using the installer

1. In your browser, download the installer bundle from the innovation lab.
    - https://store.servicenow.com/sn_appstore_store.do#!/store/innovationlab
2. Double-click the zip file to extract the OS-specific installers.
3. Locate and right-click the `snc-1.0.0-osx-installer` file. If prompted, select Open to trust and launch the installer.
4. Follow the on-screen instructions
    - When prompted, ensure the 'Add to PATH' option is selected (requires `sudo`).
    - If you do not have `sudo` access, you will have to manually create a symlink file in your `$PATH` that points to the `snc` program by using the following commands at the command prompt. If your `$PATH` includes a folder you can write to, you can run the following command without `sudo` if you specify that folder as the target's path. If you don't have a writable folder in your `$PATH`, you must use `sudo` in the command to get permissions to write to the specified target folder. The default location for a symlink is `/usr/local/bin/`.

      ```
      $ sudo ln -s /folder/installed/ServiceNow CLI/bin/snc /usr/local/bin/snc
      ```

5. To verify that the ServiceNow CLI is installed and the shell can find and run the `snc` command in your `$PATH`, use the following commands.

   ```
   $ which snc
   /usr/local/bin/snc
   $ snc version
   ```

```
{
   "extensions": {},
   "snc": "1.0.0"
}
```

Windows

Install the ServiceNow CLI on Windows.

Prerequisites

Before you can install the ServiceNow CLI on Windows, be sure you have the following:

- A 64-bit version of Windows 10 or later
- Admin rights to install software

Install the ServiceNow CLI on Windows using the installer

1. In your browser, download the installer bundle from the innovation lab.
    - https://store.servicenow.com/sn_appstore_store.do#!/store/innovationlab
2. Unzip the downloaded bundle file to extract the OS-specific installers
3. Locate and right-click the `snc-1.0.0-windows-x64-installer.exe` file. If prompted, select Yes to trust and launch the installer.
4. Follow the on-screen instructions. By default, the ServiceNow CLI installs to `C:\Program Files\ServiceNow CLI`.
5. To confirm the installation, open the **Start** menu, search for cmd to open a command prompt window. At the command prompt use the `snc version` command.

    ```
    C:\> snc version
    {
       "extensions": {},
       "snc": "1.0.0"
    }
    ```

    If Windows is unable to find the program, you might need to close and reopen the command prompt window to refresh the path.

Linux

Install the ServiceNow CLI on Linux.

Prerequisites

- You must be able to extract or "unzip" the downloaded package. If your operating system doesn't have the built-in unzip command, use an equivalent.
- The ServiceNow CLI supports 64-bit versions of recent distributions of CentOS and Ubuntu.

Install the ServiceNow CLI on Linux using the installer

1. In your browser, download the installer bundle from the innovation lab.
    - https://store.servicenow.com/sn_appstore_store.do#!/store/innovationlab
2. Unzip the downloaded bundle file to extract the OS-specific installers.
3. Locate and run the file `snc-1.0.0-linux-x64-installer.run`.
4. Follow the on-screen instructions.
    - When prompted, ensure the 'Add to PATH' option is selected (requires `sudo`).
    - If you do not have `sudo` access, you will have to manually create a symlink file in your `$PATH` that points to the `snc` program by using the following commands at the command prompt. If your `$PATH` includes a folder you can write to, you can run the following command without `sudo` if you specify that folder as the target's path. If you don't have a writable folder in your `$PATH`, you must use `sudo` in the command to get permissions to write to the specified target folder. The default location for a symlink is `/usr/local/bin/`.

      ```
      $ sudo ln -s /folder/installed/ServiceNow CLI/bin/snc /usr/local/bin/snc
      ```
5. To verify that the ServiceNow CLI is installed and the shell can find and run the `snc` command in your `$PATH`, use the following commands.

```
$ which snc
~/ServiceNow CLI/bin/snc
$ snc version
{
   "extensions": {},
   "snc": "1.0.0"
}
```

Configuring the CLI

Configure the settings that the ServiceNow CLI uses to interact with an instance.

Initial configuration

In order to enable the ServiceNow CLI to communicate with an instance, set up a default profile using the `configure profile set` command. The CLI prompts you for the following pieces of information:

Host

The host name of the ServiceNow instance to connect to. Supports both the full URL (https://myinstance.service-now.com) or just the hostname (myinstance).

Login method

The login method to use to connect to the instance. Supports basic, OAuth, and OAuth + MFA.

Username

The username to use to connect to the instance.

Password

The password to use to connect to the instance.

Client id

When login method is OAuth or OAuth+MFA, this is the client ID to use to connect to the instance.

Client secret

When login method is OAuth or OAuth+MFA, this is the client secret to use to connect to the instance.

Authentication code

When login method is OAuth+MFA, this is the authentication code to use to connect to the instance.

Default output format

Specifies how the command results are formatted. Options are JSON, YAML, text, and table.

The ServiceNow CLI stores the `credentials` in the host system's keychain while the non-sensitive information is stored as a *profile* named `default` in the CLI's configuration file. By default, the information in this profile is used when you run a ServiceNow CLI command that does not explicitly specify a profile to use.

The following example shows sample values. Replace them with your own.

```
$ snc configure profile set
Host: myinstance.service-now.com
Login Method (Basic,OAuth,OAuth+MFA): Basic
Username: myusername
Password: mypassword
```

Configuration file

The ServiceNow CLI stores profile information in a `config.json` file which, by default, is located in the user's home directory (`~/.snc/config.json`). The file is formatted as JSON with each profile having its own node. By default, the ServiceNow CLI uses the settings found in the profile named `default`. To use alternate settings, you can create and reference additional profiles. For more information, see Named profiles.

Named profiles

A *named profile* is a collection of settings and credentials that you can apply to a ServiceNow CLI command. When you specify a profile to run a command, the settings and credentials are

used to run that command. You can specify one profile that is the default and is used when no profile is explicitly referenced. Other profiles have names that you can specify as a parameter on the command line for individual commands.

The ServiceNow CLI supports using any number of multiple *named profiles* that are stored in the config file and the host system's keychain. You can configure additional profiles by using the `snc configure profile set` command with the `--profile` option.

The following example shows a config file with two profiles. The first, `default`, is used when you run a CLI command with no profile. The second is used when you run a CLI command with the `--profile` user1 parameter. Each profile can use different credentials and even specify different hosts and output formats.

`~/.snc/config.json` (Linux & Mac) or `%USERPROFILE%\.snc\config.json` (Windows)

```
{
   "profiles":{
      "default":{
         "host":"https://myinstance.service-now.com",
         "loginmethod":"basic",
         "username":"admin",
         "output":"json",
         "hostversion":"Paris",
         "appversion":"1.0"
      },
      "user1":{
         "host":"https://otherinstance.service-now.com",
         "loginmethod":"basic",
         "username":"user1",
         "output":"yaml",
         "hostversion":"Paris",
         "appversion":"1.0"
      }
   }
}
```

To use a named profile, add the --profile *profile-name* option to your command.

```
$ snc record create --profile user1
```

To see a list of configured profiles, use the list command.

```
$ snc configure profile list
{
   "profiles":{
      "default":{
         "host":"https://myinstance.service-now.com",
         "loginmethod":"basic",
         "username":"admin",
         "output":"json",
```

```
        "hostversion":"Paris",
        "appversion":"1.0"
    }
}
```

Command line options

You can use the following command line options to override the default configuration settings or any corresponding profile setting when executing a command. You can't use command line options to directly specify credentials, although you can specify which profile to use.

--help

Provides help information for the specified command. This includes the description, supported arguments and examples of usage.

--debug

A boolean that provides debug logging output when executing a command. In addition to the debug option, the ServiceNow CLI logs all command executions to a log file located in `~/.snc/.logs` (Linux & Mac) or `%USERPROFILE%\.snc\.logs` (Windows).

--profile

Specifies the *named profile* to use for this command. To set up additional named profiles, you can use the `snc configure profile set` command with the `--profile` option.

```
$ snc configure profile set --profile <profilename>
```

--output

Specifies the output format to use for this command. The ServiceNow CLI supports the following output formats.

- **json**: The output is formatted as JSON.
- **yaml**: The output is formatted as YAML.
- **text**: The output is formatted as multiple lines of tab-separated string values.
- **table**: The output is formatted as a table which presents the information in a human-friendly format.

Using the CLI

Getting help

To get help with any command, type `--help` at the end of the command name.

For example, the following command displays help for the general ServiceNow CLI options and the available top-level commands.

```
$ snc --help
```

The following command displays the available profile specific commands.

```
$ snc configure profile --help
```

The help for each command is divided into eight sections:

Name

The name of the command.

```
Name
     set
```

Description

The description of what the command does and its return values.

```
Description
     Configure connection profiles. This command is interactive and will prompt you for
     each configuration value.
```

Synopsis

The basic syntax for using the command and its options.

```
Synopsis
     snc configure profile set [arguments]
```

Available commands

The commands available under the current command group.

```
Available Commands
 list     : Lists the configured connection profiles.
 refresh  : Updates the available commands from the instance for the given profile.
 remove   : Removes the specified connection profile.
 set      : Configures connection profiles in order to communicate with an instance.
```

Command groups

The command groups available under the current command group.

```
Command Groups
    profile  : Set, view, and remove connection profiles.
```

Arguments

A description of each of the arguments the command accepts.

```
Arguments
 -p, --profile string : Use a specific connection profile when executing a command.
```

Global arguments

The global arguments that the command accepts.

```
Global Arguments
    -d, --debug           Print logs to console.
    -h, --help            Display detailed help information.
    -o, --output string   Set the format for printing command output.
```

Examples

Examples showing the usage of the command and its options.

```
Examples
  Create a new profile to save as the default:
    $ snc configure profile set
      Host:
      Login method:
      Username:
      Password:
      Client id:
      Client secret:
```

Command structure

The ServiceNow CLI uses a hierarchical structure on the command line that must be specified in this order:

1. The base call to the snc program.
2. The top-level command group followed by any nested command group(s).
3. The subcommand that specifies which operation to perform.
4. General CLI arguments required by the operation. You can specify these in any order as long as they follow the first three parts.

```
$ snc <command-group> <subcommand> [arguments]
```

Arguments can take various types of input values, such as numbers, strings, and JSON structures. What is supported is dependent upon the command and subcommand you specify.

Specifying parameter values

Many parameters used in the ServiceNow CLI are simple string or numeric values, such as the `table` and `table name` in the following example.

```
$ snc record create --table incident --data "{short_description: 'New Incident'}"
```

You can surround strings that do not contain any space characters with quotation marks or not. However, you must use quotation marks around strings that include one or more space characters.

Controlling command output

The ServiceNow CLI supports four output formats:

- **json**: The output is formatted as JSON.
- **yaml**: The output is formatted as YAML.
- **text**: The output is formatted as multiple lines of tab-separated string values.
- **Table:** The output is formatted as a table which presents the information in a human-friendly format.

Command output can be specified in two ways:

- **Using the `output` option in a named profile in the `config` file:** The following example sets the default output format to `text`.

```
{
   "profiles":{
      "default":{
         "output":"text"
      }
}
```

- **Using the `--output` argument on the command line:** The following example sets the output of only this one command to `json`. Using this option on the command overrides any currently set value in the `config` file.

```
$ snc record query --table incident --query 'active=true' --output json
```

JSON output format

JSON is the default output format of the ServiceNow CLI. Most programming languages can easily decode JSON strings using built-in functions or with publicly available libraries.

The following is an example of JSON output.

```
$ snc configure profile list --output json
{
   "default": {
      "appversion": "1.0.8",
      "host": "https://snccli.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "json",
      "username": "admin"
   },
   "user1": {
      "appversion": "1.0.8",
      "host": "https://snccli.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "yaml",
      "username": "admin"
   }
}
```

YAML output format

YAML is a good choice for handling the output programmatically with services and tools that emit or consume YAML-formatted strings.

The following is an example of YAML output.

```
$ snc configure profile list --output yaml
default:
  appversion: 1.0.8
  host: https://snccli.service-now.com
  hostversion: Paris
  loginmethod: basic
  output: json
  username: admin
user1:
  appversion: 1.0.8
  host: https://snccli.service-now.com
  hostversion: Paris
  loginmethod: basic
  output: yaml
  username: admin
```

Text output format

The text format organizes the ServiceNow CLI output into tab-delimited lines. It works well with traditional Unix text tools such as grep, sed, and awk, and the text processing performed by PowerShell.

The following is an example of text output. Each field is tab separated from the others, with an extra tab where there is an empty field.

```
$ snc configure profile list --output text
default  https://snccli.service-now.com  Paris  1.0.8  basic  admin  json
user1    https://snccli.service-now.com  Paris  1.0.8  basic  admin  yaml
```

Table output format

The table format produces human-readable representations of complex ServiceNow CLI output in a tabular form.

```
$ snc configure profile list --output table
NAME     HOST     HOST VERSION  APP VERSION  LOGIN METHOD  USERNAME  OUTPUT
-------------------------------------------------------------------------------
default  snccli   Paris         1.0.8        basic         admin     json
user1    snccli   Paris         1.0.8        basic         admin     yaml
```

Extensions

The ServiceNow CLI offers the capability to load extensions. Extensions provide additional functionality that is not shipped as part of the CLI but run as CLI commands.

Find extensions

To see the available extensions provided by ServiceNow, use the `snc extension list-available` command.

```
$ snc extension list-available -o table
NAME          DESCRIPTION                                  VERSION          INSTALLED
-------------------------------------------------------------------------------------
ui-component  Build and deploy Now Experience Components   19.0.0-alpha.15  false
```

Install extensions

Once you have found an extension to install, use the `snc extension add` command.

```
$ snc extension add --name <extension-name>
```

Update extensions

Update an extension using the `snc extension update` command.

```
$ snc extension update --name <extension-name>
```

Uninstall extensions

If an extension is no longer needed, remove it using the `snc extension remove` command.

```
$ snc extension remove --name <extension-name>
```

UI-Component extension

The `ui-component` extension provides the capability to build and deploy Now Experience components.

Prerequisites

- The ui-component extension is compatible with Windows 10 and MacOS (Yosemite and later).
- Node version `12.16.1` or greater. For instructions on how to install node, please visit here.
- npm version `6.13.14` or greater. For instructions on how to install npm, please visit here.

Installing the extension

To install the ui-component extension, run the following command.

```
$ snc extension add --name ui-component
```

Verify the installation by running the extension with the `--help` argument.

```
$ snc ui-component --help
```

Creating a component project

The first step in creating a component is to scaffold a component project. This will create all the necessary files to start developing a component. **Note**: The project command can only be run within an empty directory.

```
$ snc ui-component project --name @myorg/movie-quotes --description 'A web component that prints movie quotes.'
```

Parameters

- **name** (required): The project name. This should be a valid and unique `npm` package name.
- **description**: The project description to be available in the `npm` registry and the *Plugins* list.
- **scope**: Suggested scope name to be assigned to this project and its components.
    - Scope must be unique.
    - If provided, scope might be validated against the current host.
    - Scope is limited to 18 characters.
    - Scope must be "snake_cased".
    - If the offline option (below) is specified, then scope is mandatory.
    - Must follow the form x_customerprefix_componentname, where:
        - customerprefix is the value in the glide.appcreator.company.code system property on your instance.
        - componentname is the value provided in the component's name parameter when you created the project.
    - if scope is not provided, the Now CLI defaults the scope name to x_{vendor_prefix}_{package_name}.
- **offline**: Does not connect to the current host during the scaffolding process, skipping validation of the given scope name. Defaults to false.

Running a project locally

To run your component project on your localhost, use the `snc ui-extension develop` command. This runs a local development server where you can quickly evaluate and iterate on your components.

```
$ snc ui-component develop
```

Parameters

- **entry**: Path to the test module. Defaults to example/index.js.

- **open**: Opens the default browser and navigates to the test page. Defaults to false.
- **port**: Port where the development server should listen for incoming connections. Defaults to 8081.
- **host**: Host address to use if you want your local dev-server to be accessible externally by others. This is typically set to 0.0.0.0.

Deploying a component to an instance

To deploy your component to your target instance, use the `snc ui-component deploy` command. This creates or updates all the necessary metadata files within the specified scope.

```
$ snc ui-component deploy
```

Parameters

- **open**: Opens the default browser and navigates to the connected instance. Defaults to false.
- **force**: Deploys component changes and overwrites existing component records. Defaults to false.

Development configuration

Component authors have the ability to override default configurations and customize their build, test, and development processes. Be careful with these configuration changes as they can greatly alter the ui-component's behavior.

Make configuration changes using the `now-cli.json` file, found at the root of each component project.

Here is a brief description of each property found in the `now-cli.json` configuration file.

```
{
   "development":{
      "proxy":{
         "origin":"https://yourinstance.com",
         "port":3000,
         "proxies":[
            "/api"
         ]
      }
   }
}
```

Parameters:

- **development.proxy.headers:** Headers to append to your proxied request.

- **development.proxy.origin:** Host to open a proxy with (your instance host address).
- **development.proxy.port:** The port the proxy will run on.
- **development.proxy.proxies:** URL glob patterns to pass-through to the proxy server.

Reference

snc configure

The configure command group lets you set, view, and remove properties used by the CLI.

Commands

snc configure profile

Set, view, and remove connection profiles. A default profile must first exist in order to communicate with a ServiceNow instance.

snc configure profile set

Configure connection profiles. This command is interactive and prompts you for each configuration value. You can configure a named profile by using the --profile argument. Configuration values are stored in a configuration file (default location is ~/.snc). When prompting for information, the CLI displays current values in brackets []. To keep an existing value, press the Enter key. **Note**: Sensitive credential information is only stored in the OS keychain.

```
$ snc configure profile set [--profile profile-name]
```

snc configure profile list

List the configured connection profiles. For each profile, the CLI displays host information, version details, username, login method, and preferred output format. It does not display sensitive information such as passwords or client IDs. To see a specific connection profile, use the --profile argument.

```
$ snc configure profile list [--profile profile-name]
```

snc configure profile refresh

Updates the available commands from the instance for the given profile. If you do not include a profile argument, the CLI uses the default profile. Use this command after modifying any of the command configurations on the corresponding instance in order to keep the CLI up to date.

```
$ snc configure profile refresh [--profile profile-name]
```

snc configure profile remove

Removes the specified connection profile. You cannot remove the default connection profile using this command. To remove the default connection profile, edit the configuration file manually.

```
$ snc configure profile remove [--profile profile-name]
```

snc extension

Manage and update extensions which provide additional functionality as CLI commands.

snc extension list-available

List available extensions.

```
$ snc extension list-available
```

snc extension list

List installed extensions.

```
$ snc extension list
```

snc extension add

Add an extension to the CLI.

```
$ snc extension add [-n, --name name-of-extension]
```

snc extension remove

Remove an extension from the CLI.

```
$ snc extension remove [-n, --name name-of-extension]
```

snc record

Perform create, read, update, and delete (CRUD) operations on existing tables. Note, these commands are only available when connected to an instance with the CLI Metadata app installed.

snc record create

Inserts one record in a specified table. You cannot insert multiple records using this command.

```
$ snc record create [--table table, --data data]
```

Arguments

- **table:** Name of the table in which to save the record.
- **data:** Field name and the associated value for each field to define in the specified record (JSON format).

snc record delete

Deletes the specified record from the specified table.

```
$ snc record delete [--table table, --sysid sys_id]
```

Arguments

- **table:** Name of the table in which to delete the record.
- **sysid:** Sys ID of the record to delete.

snc record get

Retrieves the record identified by the specified Sys ID from the specified table.

```
$ snc record get [--table table, --sysid sys_id]
```

Arguments

- **table:** Name of the table from which to retrieve the record.
- **sysid:** Unique identifier of the record to retrieve.

snc record query

Retrieves multiple records from a specified table.

```
$ snc extension query [--displayvalue displayValue, --fields fields, --limit limit, --
offset offset, --query query, --table table]
```

Arguments

- **table**: Name of the table in which to query the records.
- **query**: Encoded query used to filter the result set. Syntax: --query '<col_name><operator><value>'.

- **fields**: Comma-separated list of field names to return from the database.
- **limit**: Maximum number of records to return.
- **offset**: Starting record index for which to begin retrieving records. Use this value to paginate record retrieval.
- **displayValue**: For reference and choice fields, determines whether to retrieve the display value or actual value from the database.

snc record update

Updates the specified record with the given data attributes.

```
$ snc record update [--sysid sys_id, --table table, --data data]
```

Arguments

- **table**: Name of the table in which to save the record.
- **sysid**: Sys ID of the record to update.
- **data**: Field name and the associated value for each field to define in the specified record (JSON format).

snc version

Show the version of the snc CLI client and any installed extensions.

```
$ snc version
```