Salesforce DX Setup Guide

Version 44.0, Winter '19

CONTENTS

CHAPTER 1   Before You Begin

Before you begin to explore using scratch orgs, second-generation packages, or Salesforce Extensions for VS Code, you must set up your environment.

1. Enable the Developer Hub (Dev Hub) in your Developer Edition, trial, or production org (if you're a customer), or your business org (if you're an AppExchange partner).

   To explore the new tooling, sign up for a Developer Edition org on Salesforce Developers.

2. Enable Second-Generation Packaging (2GP) in your org so you can develop 2GP packages.

3. If you want to include your team, you can add users to your Dev Hub org.

4. Ensure that your computers meet all system requirements.

5. If you want to evaluate Salesforce Extensions for VS Code, download and install the extension pack.

6. Download the sample projects from GitHub.

CHAPTER 2 System Requirements

Use this list of system requirements to get the most out of the Salesforce CLI and developer tools.

Operating Systems

The Salesforce CLI supports the following operating systems.

- Windows—Windows 7 (64-bit and 32-bit) or later
- Mac—macOS 10.11 or later
- Linux—Ubuntu 14.0.4

Code Editor or IDE

You can use any code editor, including Salesforce Extensions for VS Code, a set of Visual Studio Code extensions that is designed for Salesforce DX.

Version Control System

You can use any version control system (VCS). We recommend that you use GitHub to take advantage of the samples in our GitHub repository.

SEE ALSO:

GitHub

CHAPTER 3   CLI and Plug-In Versions

Salesforce supports only the most current versions of the Salesforce CLI and the `salesforcedx` plug-in.

To upgrade to the current version, run `sfdx update`.

By default, the CLI periodically checks for and installs updates. To disable auto-update, set the SFDX_AUTOUPDATE_DISABLE environment variable to `true`.

To check the version of the Salesforce CLI installed on your computer, run `sfdx version`. To check the version of the installed `salesforcedx` plug-in, run `sfdx plugins --core`.

SEE ALSO:

Update the CLI

Disable Automatic Update of the CLI and Plug-In

CHAPTER 4   Enable Dev Hub in Your Org

Enable Dev Hub in your org so you can create and manage scratch orgs from the command line. Scratch orgs are disposable Salesforce orgs to support development and testing.

Enabling Dev Hub in a production or business org is completely safe and won't cause any performance or customer issues. Dev Hub comprises objects with permissions that allow admins to control the level of access available to a user and an org.

Consider these factors if you select a trial or Developer Edition org as your Dev Hub.

- You can create up to 6 scratch orgs and package versions per day, with a maximum of 3 active scratch orgs.
- Trial orgs expire on their expiration date.
- Developer Edition orgs that are inactive for 365 days are deactivated.

- You can define a namespace in a Developer Edition org that isn't your Dev Hub, and you can enable Dev Hub in a Developer Edition org that doesn't contain a namespace.
- If you plan to create package versions or run continuous integration jobs, it's better to use a production or business org as your Dev Hub because of higher scratch org and package version limits. Package versions are associated with your Dev Hub org. When a trial or Developer Edition org expires, you lose access to the package versions.

> 📝 Note: Partner trial orgs signed up from the partner community have different scratch org limits. See Scratch Org Allocations for Partners. Partners can create partner edition scratch orgs: Partner Developer, Partner Enterprise, Partner Group, and Partner Professional. This feature is available only if creating scratch orgs from a Dev Hub in a partner business org. See Supported Scratch Org Editions for Partners in the *ISVforce Guide* for details.

The Dev Hub org instance determines where scratch orgs are created.

- Scratch orgs created from a Dev Hub org in Government Cloud are created on a Government Cloud instance.
- Scratch orgs created from a Dev Hub org in Public Cloud are created on a Public Cloud instance.

To enable Dev Hub in an org:

1. Log in as System Administrator to your Developer Edition, trial, or production org (if you're a customer), or your business org (if you're an ISV).

2. From Setup, enter `Dev Hub` in the Quick Find box and select **Dev Hub**.

   If you don't see Dev Hub in the Setup menu, make sure your org is one of the supported editions.

3. To enable Dev Hub, click **Enable**.

After you enable Dev Hub, you can't disable it.

SEE ALSO:

Salesforce Help: Deactivate an Org

CHAPTER 5 Enable Second-Generation Packaging

Enable Second-Generation Packaging (2GP) in your org so you can develop unlocked packages (generally available) or managed packages (beta). You can work with the packages in scratch orgs, sandbox orgs, and target subscriber orgs.

Enable Dev Hub in your org.

1. Log in to the org where you've enabled Dev Hub.

2. From Setup, enter `Dev Hub` in the Quick Find box and select **Dev Hub**.

3. In the section for Second-Generation Packaging, click **Non-GA Service Agreement** to read the service agreement.

4. Click **Enable Unlocked Packages (GA) and Second-Generation Managed Packages (Beta)**.

   After you enable Second-Generation Packaging, you can't disable it.

5. (Optional) Allow non-admin users to access the Dev Hub to create packages.

   Assign non-admin users the Create and Update Second Generation Packages user permission. See Add Salesforce DX Users for details.

CHAPTER 6   Add Salesforce DX Users

System administrators can access the Dev Hub org by default. You can enable more users to access the Dev Hub org so that they can also create scratch orgs.

You can use Salesforce DX with these Standard user licenses: Salesforce, Salesforce Platform, and Salesforce Limited Access - Free (partners only).

You can add a user with the System Administrator profile. You can also add a user with a Standard User profile as long as you apply the set of permissions required for Salesforce DX. Because you're adding users to a Dev Hub org, avoid adding them as system administrators unless their work requires that level of authority.

1. Add the user to your Dev Hub org, if necessary.

   a. In Setup, enter `Users` in the Quick Find box, then select **Users**.

   b. Click **New User**.

   c. Fill out the form, and assign the System Administrator or Standard User profile.

   d. Click **Save**.

   If you're adding a System Administrator user, you can stop here.

2. If you're adding a Standard User, create a permission set for Salesforce DX users if you don't have one.

   a. From Setup, enter `Permission Sets` in the Quick Find box, then select **Permission Sets**.

   b. Click **New**.

   c. Enter a label, API name, and description. The API name is a unique name used by the API and managed packages.

   d. Select a user license option. If you plan to assign this permission set to multiple users with different licenses, select **None**.

   e. Click **Save**. The permission set overview page appears. From here, you can navigate to the permissions you want to add or change for Salesforce DX. For the required permissions, see Permission Set for Salesforce DX Users on page 8.

3. Apply the Salesforce DX permission set to the Standard User.

   a. From Setup, enter `Permission Sets` in the Quick Find box, then select **Permission Sets**.

   b. Select the Salesforce DX permission set.

   c. In the permission set toolbar, click **Manage Assignments**.

   d. Click **Add Assignments**.

   e. Select the user to assign the permission set to.

   f. Click **Assign**.

g. Click **Done**.

You can limit a user's access by modifying the permissions.

Permission Set for Salesforce DX Users

To give full access to the Dev Hub org, the permission set must contain these permissions.

- Object Settings > Scratch Org Info > Read, Create, and Delete
- Object Settings > Active Scratch Org > Read and Delete
- Object Settings > Namespace Registry > Read

To work with second-generation packages in the Dev Hub org, the permission set must also contain:

- System Permissions > Create and Update Second-Generation Packages

This permission provides access to:

Salesforce CLI Command	Tooling API Object (Create and Edit)
`force:package:create`	Package2
`force:package:version:create`	Package2VersionCreateRequest
`force:package:version:update`	Package2Version

SEE ALSO:

Permission Sets (*Salesforce Help*)

Salesforce DX Developer Guide

CHAPTER 7   Install the Salesforce CLI

Use the CLI commands to create environments for development and testing, synchronize source code between your scratch orgs and version control system, and execute test suites.

Install the CLI on macOS

You install the Salesforce CLI on macOS with a `.pkg` file.

1. [Download](#) the `.pkg` file.

2. Double-click the `.pkg` file.

SEE ALSO:

[Verify Your Installation](#)

[Disable Automatic Update of the CLI and Plug-In](#)

Install the CLI on Windows

You install the Salesforce CLI on Windows with an `.exe` file.

[Download](#) and run the Windows installer.

> ⚠ **Warning:** The Salesforce CLI works best within the native Windows command prompt (`cmd.exe`) and the Microsoft Powershell. We do not recommend using the Salesforce CLI with a Linux terminal emulator, such as Windows 10 Subsystem for Linux, cygwin, or MinGW, because support for bugs is limited.

SEE ALSO:

[Verify Your Installation](#)

[Disable Automatic Update of the CLI and Plug-In](#)

Install the CLI on Linux

The Linux version of the Salesforce CLI is distributed as a tarball.

Find the download URL for your tarball from [this manifest file](#). The `downloads` section lists unversioned URLs for the latest installers, which are especially useful for CI use cases. Select the URL based on your target platform.

1. Download or `wget` one of these tarballs.

```
wget https://developer.salesforce.com/media/salesforce-cli/sfdx-linux-amd64.tar.xz
```

```
wget https://developer.salesforce.com/media/salesforce-cli/sfdx-linux-386.tar.xz
```

2. Create an `sfdx` directory.

```
mkdir sfdx
```

3. Unpack the contents for your tarball version:

```
tar xJf sfdx-linux-amd64.tar.xz -C sfdx --strip-components 1
```

```
tar xJf sfdx-linux-386.tar.xz -C sfdx --strip-components 1
```

`-C` unpacks the contents in the `sfdx` directory, while `--strip-components 1` removes the root path component.

4. Run the install script.

```
./sfdx/install
```

The Salesforce CLI is installed in `/usr/local/bin/sfdx`.

SEE ALSO:

    Verify Your Installation

    Disable Automatic Update of the CLI and Plug-In

Install the CLI with `npm`

If you've installed `Node.js` on your computer, you can use `npm` to install the Salesforce CLI. This method lets you install the Salesforce CLI from the command line and can be especially useful for continuous integration (CI) use cases.

This installation method is a good option if you don't have administrator permissions on your workstation, or if group policy blocks CLI installation and updates. Installing the CLI with `npm` doesn't require root permissions.

1. Ensure that the long-term support (LTS) version of Node.js is installed on your computer. To find the LTS version number, check https://nodejs.org/en/download/. To check your version number, run:

```
node --version
```

2. Run this command.

```
npm install sfdx-cli --global
```

If you receive a permission error when installing the CLI using npm, we recommend not using sudo. See Fixing npm permissions.

SEE ALSO:

    Verify Your Installation

    npm Documentation

Verify Your Installation

Verify your Salesforce CLI installation and plug-in versions.

Run this command to verify the Salesforce CLI version:

```
sfdx --version
sfdx-cli/6.0.10-3713d7b alpha (darwin-x64) node-v8.6.0
```

Run this command to verify the Salesforce CLI plug-in version:

```
sfdx plugins --core
salesforcedx 41.2.0 (core)
```

This command returns a list of the other plug-ins installed in the CLI:

```
sfdx plugins
```

The core `salesforcedx` plug-in is not included in the preceding list unless you've explicitly installed a newer version with the `sfdx plugins:install` command.

Run this command to return a list of the command families in the `force` topic:

```
sfdx force --help
```

This command returns all the `force` commands:

```
sfdx force:doc:commands:list
```

SEE ALSO:

Install a Specific Version of the `salesforcedx` Plug-In

By default, the latest version of the Salesforce CLI and `salesforcedx` plug-in is installed when you install the CLI for the first time. Sometimes, however, you want to use a specific version of the `salesforcedx` plug-in.

Run this command to install a specific version of the Salesforce CLI, in this example, version 42.2.0:

```
sfdx plugins:install salesforcedx@42.2.0
```

CHAPTER 8   Use the Salesforce CLI from Behind a Company Firewall or Web Proxy

If you install or update the Salesforce CLI on a computer that's behind a company firewall or web proxy, you sometimes receive error messages. In this case, you must further configure your system.

You get an error similar to the following when you run a command after installing the CLI binary behind a firewall or web proxy. This error is from a Linux computer, but Windows and macOS users sometimes see a similar error.

```
sfdx-cli: Updating CLI... !
 □  'ECONNRESET': tunneling socket could not be established,
cause=connect EHOSTUNREACH 0.0.23.221:8080 - Local
(10.126.148.39:53107)
```

To address this issue, run these commands from your terminal or Windows command prompt, replacing *username:pwd* with your web proxy username and password. If your proxy doesn't require these values, omit them. Also replace *proxy.company.com:8080* with the URL and port of your company proxy.

```
npm config set https-proxy https://username:pwd@proxy.company.com:8080
npm config set proxy https://username:pwd@proxy.company.com:8080
```

Then set the HTTP_PROXY or HTTPS_PROXY environment variable to the full URL of the proxy. For example, on UNIX:

```
export HTTP_PROXY=https://username:pwd@proxy.company.com:8080
```

```
export HTTPS_PROXY=https://username:pwd@proxy.company.com:8080
```

On a Windows machine:

```
set HTTP_PROXY=https://username:pwd@proxy.company.com:8080
```

```
set HTTPS_PROXY=https://username:pwd@proxy.company.com:8080
```

If You Still See an Error

If you set the proxy environment variable, and you still see error messages, it's possible that your proxy requires an extra certificate authority (CA). Ask your IT department where to find or download the certificates.

Set this environment variable to point to the CA file: NODE_EXTRA_CA_CERTS.

CHAPTER 9   Update the CLI

If you want to ensure that you are running the latest version of Salesforce CLI, you can manually update
it.

If You Installed the Salesforce CLI Using the Installer

To install the latest Salesforce CLI version, run:

```
sfdx update
sfdx-cli: Updating CLI from 6.0.0-0743bea5 to 6.0.0-aebbfd66
(alpha)... done
sfdx-cli: Updating CLI... already on latest version: 6.0.0-aebbfd66
sfdx-cli: Updating plugins... done
```

By default, the CLI periodically checks for and installs updates. To disable auto-update, set the
SFDX_AUTOUPDATE_DISABLE environment variable to true.

If You Installed the Salesforce CLI Using npm

The auto-update option is not available. To update the Salesforce CLI using npm, run:

```
npm install --global sfdx-cli
```

Windows: Improve CLI Update Speed

Anti-virus software, such as Windows Defender, slows a CLI update because it scans all the files as they
are downloaded. To improve the update speed, add the %LOCALAPPDATA%\sfdx directory to the
list of directories that are excluded from the scan. We recommend that you consult your security
administrator before making this configuration change.

Disable Automatic Update of the CLI and Plug-In

When you run a command, the Salesforce CLI checks to see if you have the latest version. If not, the CLI automatically updates itself and the `salesforcedx` plug-in. You can disable this automatic update with an environment variable.

To remain on the current version of the CLI and disable automatic updates, set the `SFDX_AUTOUPDATE_DISABLE` environment variable to `true`. For example, on Unix:

```
export SFDX_AUTOUPDATE_DISABLE=true
```

CHAPTER 10 Uninstall the CLI Binary or Plug-In

Uninstalling the CLI removes it entirely from your computer.

macOS or Linux

Enter all these commands in a terminal:

```
sudo rm -rf /usr/local/sfdx
sudo rm -rf /usr/local/lib/sfdx
sudo rm -rf /usr/local/bin/sfdx
sudo rm -rf ~/.local/share/sfdx ~/.config/sfdx ~/.cache/sfdx
sudo rm -rf ~/Library/Caches/sfdx
```

Windows

1. Select **Start > Control Panel > Programs > Programs and Features**.

2. Select **SFDX CLI**, and click **Uninstall**.

3. Inside your home directory, delete the `.config\sfdx` directory.

If the CLI is still installed, delete the `%LOCALAPPDATA%\sfdx` directory in Program Files.

Uninstall the `salesforcedx` Plug-In

Enter this command from a terminal or Windows command prompt:

```
sfdx plugins:uninstall salesforcedx
```

CHAPTER 11 Sample Source in GitHub

While you're exploring Salesforce DX, look at some sample repos we've provided on GitHub to help you understand the concepts.

- `sfdx-simple`—A sample with two Visualforce pages, a controller, and Apex tests.
- `dreamhouse-sfdx`—A more complex sample app with multiple Apex classes, configuration items, Lightning components, Visualforce components, and custom objects. The `dreamhouse-sfdx` sample is a standalone application.
- `sfdx-travisci`-A continuous integration (CI) sample that shows how to use Salesforce DX with Travis CI. Travis CI is a cloud-based continuous integration (CI) service for building and testing software projects hosted on GitHub.

SEE ALSO:

sfdx-simple GitHub Sample

dreamhouse-sfdx GitHub Sample

sfdx-travisci GitHub Sample

CHAPTER 12 Next Steps

Read on to learn what to do after you've installed the Salesforce CLI.

For a hands-on exploration of how the Salesforce DX components work together using the DreamHouse sample, see the DreamHouse Readme.

To learn more about how to use Salesforce DX, see the *Salesforce DX Developer Guide*.

For the complete list of CLI commands and how to use them, see the *Salesforce CLI Command Reference*.

SEE ALSO:

dreamhouse-sfdx GitHub Sample Readme

Salesforce CLI Command Reference

Salesforce DX Developer Guide