Salesforce Commerce Plug-In CLI Command Reference

Salesforce, Summer '25

Last updated: July 25, 2025

CONTENTS

SALESFORCE COMMERCE PLUG-IN CLI COMMAND REFERENCE

The command reference contains information about the Salesforce CLI commands for the Commerce plug-in and their parameters.

This version of the command reference includes details about version 7.159.0 of the Salesforce CLI and the following plug-in versions:

- `@salesforce/commerce` version 238.0.15

For information about installing Salesforce CLI, see the *Salesforce CLI Setup Guide*.

For information about Salesforce CLI changes, see the *Salesforce CLI Release Notes*.

`commerce` Namespace

Commands for 1commerce developers.

`examples` Commands

`commerce:examples:convert`

Convert repo examples to SFDX scratch org.

Examples for `commerce:examples:convert`

sfdx commerce:examples:convert -f store-scratch-def.json

Command Syntax

`sfdx commerce:examples:convert`

[--json]

[--loglevel LOGLEVEL]

[-f DEFINITIONFILE]

[-d OUTPUTDIR]

[-p SOURCEPATH]

[-o TYPE]

[-y]

[-n STORE-NAME]

Parameters

`--json`

Optional

Format output as JSON.

Type: boolean

`--loglevel LOGLEVEL`

Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

`-f | --definitionfile DEFINITIONFILE`

Optional

Config file.

Type: filepath

Default value: ~/.commerce/config/store-scratch-def.json

`-d | --outputdir OUTPUTDIR`

Optional

Directory to output the conversion.

Type: string

Default value: ~/.commerce/force-app

`-p | --sourcepath SOURCEPATH`

Optional

Files to convert.

Type: string

-o | --type TYPE
Optional

The type of store you want to create.

Type: string

Permissible values are: b2c, b2b

Default value: b2c

-y | --prompt
Optional

If there is a file difference detected, prompt before overwriting file.

Type: boolean

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

`files` Commands

commerce:files:copy
Copy files from source to destination folder .commerce.

`commerce:files:copy`

Copy files from source to destination folder .commerce.

Examples for `commerce:files:copy`

sfdx commerce:files:copy -y --copySourcePath "~/myexamplefilesdirectory" --filestocopy "file1.txt,file2.txt" --dirstocopy "dir1,dir2,dir3"

sfdx commerce:files:copy --prompt --copySourcePath "~/myexamplefilesdirectory" --filestocopy "file1.txt,file2.txt" --dirstocopy "dir1,dir2,dir3"

sfdx commerce:files:copy --copySourcePath "~/myexamplefilesdirectory" --filestocopy "file1.txt,file2.txt" --dirstocopy "dir1,dir2,dir3"

Command Syntax

sfdx commerce:files:copy
[--json]

[--loglevel LOGLEVEL]

[-y]

--filestocopy FILESTOCOPY

--dirstocopy DIRSTOCOPY

```
--copysourcepath COPYSOURCEPATH
```

Parameters

--json
Optional

Format output as JSON.

Type: boolean

--loglevel LOGLEVEL
Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-y | --prompt
Optional

If there is a file difference detected, prompt before overwriting file.

Type: boolean

--filestocopy FILESTOCOPY
Required

Array of individual files to copy located directly in source directory.

Type: array

--dirstocopy DIRSTOCOPY
Required

Array of directories (including their contents) located in source directory to copy.

Type: array

--copysourcepath COPYSOURCEPATH
Required

Base path for files and directories to be copied from.

Type: string

`payments` Commands

commerce:payments:quickstart:setup
Set up a new Payment Gateway.

`commerce:payments:quickstart:setup`

Set up a new Payment Gateway.

Examples for `commerce:payments:quickstart:setup`

sfdx commerce:payments:quickstart:setup -p Stripe

Command Syntax

`sfdx commerce:payments:quickstart:setup`

[--json]

[--loglevel LOGLEVEL]

[-u TARGETUSERNAME]

[--apiversion APIVERSION]

[-p PAYMENT-ADAPTER]

[-n STORE-NAME]

Parameters

`--json`

Optional

Format output as JSON.

Type: boolean

`--loglevel LOGLEVEL`

Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

`-u | --targetusername TARGETUSERNAME`

Optional

A username or alias for the target org. Overrides the default target org.

Type: string

`--apiversion APIVERSION`

Optional

Override the API version used for API requests made by this command.

Type: string

`-p | --payment-adapter PAYMENT-ADAPTER`

Optional

Payment Adapter.

Type: string

Default value: Stripe

`-n | --store-name STORE-NAME`

Required

Name of the site to create.

Type: string

Default value: 1commerce

`products` Commands

commerce:products:import

Prepare product data files for import.

`commerce:products:import`

Prepare product data files for import.

Examples for `commerce:products:import`

sfdx commerce:products:import --store-name test-store

Command Syntax

`sfdx commerce:products:import`

[--json]

[--loglevel LOGLEVEL]

[-v TARGETDEVHUBUSERNAME]

[-u TARGETUSERNAME]

[--apiversion APIVERSION]

[-c PRODUCTS-FILE-CSV]

[-f DEFINITIONFILE]

[-o TYPE]

[-n STORE-NAME]

[-y]

Parameters

`--json`

Optional

Format output as JSON.

Type: boolean

`--loglevel LOGLEVEL`

Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-v | --targetdevhubusername TARGETDEVHUBUSERNAME
Optional

A username or alias for the target Dev Hub org. Overrides the default Dev Hub org.

Type: string

-u | --targetusername TARGETUSERNAME
Optional

A username or alias for the target org. Overrides the default target org.

Type: string

--apiversion APIVERSION
Optional

Override the API version used for API requests made by this command.

Type: string

-c | --products-file-csv PRODUCTS-FILE-CSV
Optional

The csv file containing products to import. Pass in empty value to do product-less import.

Type: string

Default value: ~/.commerce/examples/csv/Alpine-small.csv

-f | --definitionfile DEFINITIONFILE
Optional

Config file.

Type: filepath

Default value: ~/.commerce/config/store-scratch-def.json

-o | --type TYPE
Optional

The type of store you want to create.

Type: string

Permissible values are: b2c, b2b

Default value: b2c

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

-y | --prompt
Optional

If there is a file difference detected, prompt before overwriting file.

Type: boolean

`store` Commands

`commerce:store:create`

Comprehensive create and set up a store. This will create your community/store push store sources, create buyer user, import products, create search index.

Examples for `commerce:store:create`

sfdx commerce:store:create --store-name test-store

Command Syntax

`sfdx commerce:store:create`

```
[--json]
[--loglevel LOGLEVEL]
[-v TARGETDEVHUBUSERNAME]
[-u TARGETUSERNAME]
[--apiversion APIVERSION]
[-f DEFINITIONFILE]
[-o TYPE]
[-y]
[-t TEMPLATENAME]
[-b BUYER-USERNAME]
[-n STORE-NAME]
```

Parameters

`--json`

Optional

Format output as JSON.

Type: boolean

--loglevel LOGLEVEL
Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-v | --targetdevhubusername TARGETDEVHUBUSERNAME
Optional

A username or alias for the target Dev Hub org. Overrides the default Dev Hub org.

Type: string

-u | --targetusername TARGETUSERNAME
Optional

A username or alias for the target org. Overrides the default target org.

Type: string

--apiversion APIVERSION
Optional

Override the API version used for API requests made by this command.

Type: string

-f | --definitionfile DEFINITIONFILE
Optional

Config file.

Type: filepath

Default value: ~/.commerce/config/store-scratch-def.json

-o | --type TYPE
Optional

The type of store you want to create.

Type: string

Permissible values are: b2c, b2b

Default value: b2c

-y | --prompt
Optional

If there is a file difference detected, prompt before overwriting file.

Type: boolean

-t | --templatename TEMPLATENAME
Optional

Template to use to create a site .

Type: string

Default value: b2c-lite-storefront

-b | --buyer-username BUYER-USERNAME
Optional

Buyer's username.

Type: string

Default value: buyer@1commerce.com

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

`commerce:store:display`

Display buyer info.

Examples for `commerce:store:display`

sfdx commerce:store:display --store-name test-store

Command Syntax

sfdx commerce:store:display
[--json]

[--loglevel LOGLEVEL]

[-v TARGETDEVHUBUSERNAME]

[-u TARGETUSERNAME]

[--apiversion APIVERSION]

[-b BUYER-USERNAME]

[-n STORE-NAME]

[-p URLPATHPREFIX]

Parameters

--json
Optional

Format output as JSON.

Type: boolean

--loglevel LOGLEVEL
Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-v | --targetdevhubusername TARGETDEVHUBUSERNAME
Optional

A username or alias for the target Dev Hub org. Overrides the default Dev Hub org.

Type: string

-u | --targetusername TARGETUSERNAME
Optional

A username or alias for the target org. Overrides the default target org.

Type: string

--apiversion APIVERSION
Optional

Override the API version used for API requests made by this command.

Type: string

-b | --buyer-username BUYER-USERNAME
Optional

Buyer's username.

Type: string

Default value: buyer@1commerce.com

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

-p | --urlpathprefix URLPATHPREFIX
Optional

Required if different from store-name URL to append to the domain created when Experiences was enabled for this org.

Type: string

`commerce:store:open`

Open store(s).

Examples for `commerce:store:open`

sfdx commerce:store:open --store-name test-store

sfdx commerce:store:open --all

Command Syntax

sfdx commerce:store:open
[--json]

```
[--loglevel LOGLEVEL]
[-v TARGETDEVHUBUSERNAME]
[-u TARGETUSERNAME]
[--apiversion APIVERSION]
[-n STORE-NAME]
[--all]
```

Parameters

--json
Optional

Format output as JSON.

Type: boolean

--loglevel LOGLEVEL
Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-v | --targetdevhubusername TARGETDEVHUBUSERNAME
Optional

A username or alias for the target Dev Hub org. Overrides the default Dev Hub org.

Type: string

-u | --targetusername TARGETUSERNAME
Optional

A username or alias for the target org. Overrides the default target org.

Type: string

--apiversion APIVERSION
Optional

Override the API version used for API requests made by this command.

Type: string

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

--all
Optional

View All stores using sfdx force:org:open _ui/networks/setup/SetupNetworksPage page.

Type: boolean

`commerce:store:quickstart:create`

Use this command to just create a community. It will use sfdx force:community:create until a community is created or failed.

Examples for `commerce:store:quickstart:create`

sfdx commerce:store:quickstart:create --templatename 'b2c-lite-storefront'

Command Syntax

`sfdx commerce:store:quickstart:create`

```
[--json]
[--loglevel LOGLEVEL]
[-u TARGETUSERNAME]
[--apiversion APIVERSION]
[-t TEMPLATENAME]
[-n STORE-NAME]
```

Parameters

`--json`

Optional

Format output as JSON.

Type: boolean

`--loglevel LOGLEVEL`

Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

`-u | --targetusername TARGETUSERNAME`

Optional

A username or alias for the target org. Overrides the default target org.

Type: string

`--apiversion APIVERSION`

Optional

Override the API version used for API requests made by this command.

Type: string

`-t | --templatename TEMPLATENAME`

Optional

Template to use to create a site .

Type: string

Default value: b2c-lite-storefront

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

commerce:store:quickstart:setup

Set up a store.

Examples for commerce:store:quickstart:setup

sfdx commerce:store:quickstart:setup --definitionfile store-scratch-def.json

Command Syntax

sfdx commerce:store:quickstart:setup
[--json]

[--loglevel LOGLEVEL]

[-v TARGETDEVHUBUSERNAME]

[-u TARGETUSERNAME]

[--apiversion APIVERSION]

[-f DEFINITIONFILE]

[-y]

[-n STORE-NAME]

Parameters

--json
Optional

Format output as JSON.

Type: boolean

--loglevel LOGLEVEL
Optional

The logging level for this command invocation. Logs are stored in $HOME/.sfdx/sfdx.log.

Type: enum

Permissible values are: trace, debug, info, warn, error, fatal, TRACE, DEBUG, INFO, WARN, ERROR, FATAL

Default value: warn

-v | --targetdevhubusername TARGETDEVHUBUSERNAME
Optional

A username or alias for the target Dev Hub org. Overrides the default Dev Hub org.

Type: string

-u | --targetusername TARGETUSERNAME
Optional

A username or alias for the target org. Overrides the default target org.

Type: string

--apiversion APIVERSION
Optional

Override the API version used for API requests made by this command.

Type: string

-f | --definitionfile DEFINITIONFILE
Optional

Config file.

Type: filepath

Default value: ~/.commerce/config/store-scratch-def.json

-y | --prompt
Optional

If there is a file difference detected, prompt before overwriting file.

Type: boolean

-n | --store-name STORE-NAME
Required

Name of the site to create.

Type: string

Default value: 1commerce

Help for Salesforce CLI Commands

The `-h | --help` parameter shows details about Salesforce CLI topics and their commands.

For namespaces, the `-h | --help` parameter lists all topics in the namespace. For example, to see names and descriptions of all topics in the `force` namespace, run `sfdx force -h`.

For topics, the `-h | --help` parameter lists the commands and their descriptions. For example, to see all commands in the `org` topic, run `sfdx force:org -h`.

For commands, adding the `-h | --help` parameter shows parameters and usage information. For example, to see help for the `org:create` command, run `sfdx force:org:create -h`.

Help for commands has four parts.

1. **Short Description of Command**

   At the top of the `--help` output (with no heading), a short description of the command is shown. For longer descriptions, see the *Salesforce CLI Command Reference*.

2. **Usage**

   The command signature on the Usage line uses the docopt format.

- All available parameters are listed. Parameters that have short names are listed using their short names.
- Parameters that take a value show the value's type (for example, `<string>`) in angle brackets immediately after the parameter's name.
- Optional parameters are in square brackets (`[ ... ]`).
- Required parameters have no annotation.
- For parameters that accept a limited set of values, the values are shown after the parameter name, separated by pipes (`--parametername value1|value2|value3`).
- Mutually exclusive options are shown in parentheses, separated by a pipe (`( ... | ... )`).

If the command takes varargs (name-value pairs that aren't parameters), the usage signature includes `name=value...`.

Tip: To see all Salesforce CLI commands, run `sfdx commands`.

3. **Options**

   The Options section lists all the command's parameters, including their short name, long name, and purpose. For parameters that accept a value, the value name is written after an equals sign (=). The equals sign is optional when you run the command—for example, you could run `sfdx force:org:create` ***`-f=config/enterprise-scratch-def.json`*** `-a TestOrg1` or `sfdx force:org:create` ***`-f config/enterprise-scratch-def.json`*** `-a TestOrg1` with the same results.

   Parameters that accept a limited list of values include the values in parentheses, with the default value indicated by an asterisk (*).

   For more information about the parameters, see the *Salesforce CLI Command Reference*.

4. **Description**

   Usage notes and examples are below the list of parameters, in the Description section. This information is also available in the *Salesforce CLI Command Reference*.

CLI Deprecation Policy

Salesforce deprecates CLI commands and flags when, for example, the underlying API changes.

The Salesforce CLI deprecation policy is:

- Salesforce announces new and upcoming deprecations of commands and flags in the weekly Salesforce CLI release notes.
- Salesforce can deprecate a command or flag at any time.
- When you run the deprecated command, Salesforce provides a deprecation warning for a minimum of 4 months.
- Salesforce removes the deprecated command or flag 4 months, or more, after the deprecation warning first appears.
- If you use a command or flag that's been deprecated but not yet removed, you get a warning message in `stderr` in the human-readable output. If you specify JSON output, the warning is presented as a property. The message includes the plugin version in which we plan to remove the command or flag. The command help also includes deprecation information when appropriate.
- When possible, Salesforce provides a functional alternative to the deprecated command or flag.
- For our policy on changes to a Salesforce CLI command's JSON response, see Support for JSON Responses.

Discover Salesforce Plugins

Check out these other plugins that work with specific Salesforce features. These plugins are created by Salesforce.

ISV Technical Enablement Plugin

The ISVTE plugin is an on-demand Technical Evangelist. It scans your package metadata and code, and provides targeted feedback to help you improve and future-proof your app. The feedback includes a detailed metadata inventory, recommendations on features or technologies to consider using, enablement resources, and installation limitations. The feedback also includes best practices, partner alerts, guidance on improving your partner Trailblazer score, and more. While it's designed for ISV and OEM partners, anyone developing on the platform can use it.

When you install the plugin, you're asked to confirm that it's unsigned. Answer `yes`. This behavior is expected.

See GitHub for documentation and more information.

CRM Analytics Plugin

CRM Analytics is a cloud-based platform for connecting data from multiple sources, creating interactive views of that data, and sharing those views in apps.

Use the CRM Analytics CLI plugin to create scratch orgs with Analytics Studio, which you can use to develop and test source code. The plugin includes commands that call a subset of the Analytics REST API endpoints to manage CRM Analytics assets programmatically. Create and iteratively develop CRM Analytics templates. Update and delete apps, dashboards, lenses, and dataflows. Use history commands to restore previous versions of dashboards and dataflows. Manage the auto-install lifecycle for embedded templated apps.

See Develop with the Analytics Plugin for the Salesforce CLI for documentation and more information.

Salesforce Code Analyzer Plugin

The Salesforce Code Analyzer plugin is a unified tool for static analysis of source code, in multiple languages (including Apex), with a consistent command-line interface and report output. We currently support the PMD rule engine, ESLint, and RetireJS.

The plugin creates "rule violations" when the scanner identifies issues. Developers use this information as feedback to fix their code. Integrate this plugin into your continuous integration (CI) solution to continually enforce the rules and ensure high-quality code.

See Salesforce Code Analyzer for documentation and more information.