OSF Builder Suite For Salesforce Commerce Cloud Data Import Plugin

OSF Builder Suite For Salesforce Commerce Cloud :: Data Import is a very easy-to-use and highly configurable Jenkins plugin that is used to automate data imports to your Salesforce Commerce Cloud continuous integration sandbox, development or staging instance.

If you have a bug to report or maybe a feature that you wish to request, please do so on GitHub, on the project's issues page.

Features

- Simple. It does one thing, and it does it well.
- Easy to install, use and keep updated.
- Easy to configure. The plugin can be configured from the Jenkins web interface.
- Support for classical mode, Jenkins Pipelines and also the new modern Blue Ocean interface.
- Super flexible. Every little thing is configurable so that the plugin can be easily adjusted to your workflow.
- Integrated with the Jenkins credentials plugin so that your credentials are safely stored encrypted.
- Support for two factor authentication
- Good documentation. Every option is documented both here on this page but also inline in Jenkins's UI by clicking the question mark icon next to the item for which you wish to display the help information.
- Support for HTTP proxy with basic or NTLM authentication.
- Free
- Open source
- Fast builds. The data is checked for changes and if nothing changed the build is skipped so that you don't import same unchanged data over and over again.

Installation

Just go to "`Manage Jenkins > Manage Plugins > Available`", search for "`OSF Builder Suite`", select **"`OSF Builder Suite For Salesforce Commerce Cloud :: Data Import"`** and click "`Download now and install after restart`" button.

Configuration

Instance Hostname	

Hostname of the SFCC instance where this build should be deployed. Examples:

`cert.staging.`***`realm.customer`***`.demandware.net`	For deployments to a staging instance that has two factor auth enabled.
`staging-`***`realm-customer`***`.demandware.net`	For deployments to a staging instance that does not have two factor auth enabled.
`development-`***`realm-customer`***`.demandware.net`	For deployments to a development instance.
`devNN-`***`realm-customer`***`.demandware.net`	For deployments to a sandbox instance.

Business Manager Credentials	- none - ⬍	🔑 Add⬍
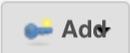
Business Manager credentials of type "`OSF Builder Suite :: Business Manager Credentials`" for the SFCC instance where this build should be deployed.

Two Factor Auth Credentials   - none - ⇕   🔑 Add

Two Factor Auth credentials of type "*OSF Builder Suite :: Two Factor Auth Credentials*" for the SFCC instance where this build should be deployed. Select *- none -* if you deploy to a instance that does not require two factor auth.

Open Commerce API Credentials   - none - ⇕   🔑 Add

Open Commerce API credentials of type "*OSF Builder Suite :: Open Commerce API Credentials*" for the SFCC instance where this build should be deployed.

Open Commerce API Version

The version to be used by the calls made to OCAPI. The Open Commerce API Version starts with the character "*v*" (lowercase) followed by the actual version number, separated by an underscore.

For example: "*v18_3*"

Archive Name

Name of the zip file that will be created by compressing your data that will be uploaded to your target instance and then imported. Only the name, without the "*.zip*" extension as it will be added by the plugin automatically.

For example: "*metadata*"

Source Path

Path (relative to the workspace) to a directory where the builder will look for data to be imported.

For example: "*scm/my-git-repo/metadata*"

Exclude Patterns   Add

You can also define a list of patterns to be ignored. If a path matches any of the patterns in this list then it will be ignored and not added to the build. The pattern needs to be relative to the source path defined above.

When a path is matched against a pattern, the following special characters can be used:

?	Matches one character (any character except path separators)
*	Matches zero or more characters (not including path separators)
**	Matches zero or more path segments

Examples:

**/*.js	Matches all .js files/dirs in a directory tree
node_modules/**	Matches the node_modules folder and all its contents
test/a??.js	Matches all files/dirs which start with an a, then two more characters and then .js, in a directory called test

`**`	Matches everything in a directory tree
`**/test/**/XYZ*`	Matches all files/dirs which start with XYZ and where there is a parent directory called test (e.g. abc\test\def\ghi\XYZ123)

Temp Directory

Path (relative to the workspace) to a temp directory, that will be used during the build. If the directory does not exist, it will be created by the builder and it will also be automatically cleaned up before each build.

For example: "`tmp/data_import`"

HTTP Proxy Host

If your Jenkins server sits behind a firewall and does not have direct access to the internet, you can specify the HTTP proxy host in this field to allow Jenkins to connect to the internet trough it.

HTTP Proxy Port

This field works in conjunction with the proxy host field to specify the HTTP proxy port.

HTTP Proxy Username

This field works in conjunction with the proxy host field to specify the username used to authenticate with the proxy.

If this proxy requires Microsoft's NTLM authentication scheme then the domain name can be encoded within the username by prefixing the domain name followed by a back-slash \ before the username, e.g "ACME\John Doe".

HTTP Proxy Password

This field works in conjunction with the proxy host field to specify the HTTP proxy password.

Disable SSL validation ☐

When this option is checked, the builder will no longer validate the SSL certificate and hostname of the target instance.

This has potential security implications so make sure you know what you are doing before enabling this option!

Open Commerce API Settings

Go to "`Administration >  Site Development >  Open Commerce API Settings`", select type "`Data`", select context "`Global`" and add following configuration:

```json
{
  "_v":"18.3",
  "clients":
  [
    {
      "client_id":"YOUR_CLIENT_ID",
      "resources":
      [
        {
          "resource_id":"/jobs/*/executions",
          "methods":["post"],
          "read_attributes":"(**)",
          "write_attributes":"(**)"
        },
        {
          "resource_id":"/jobs/*/executions/*",
          "methods":["get"],
          "read_attributes":"(**)",
          "write_attributes":"(**)"
        }
      ]
    }
  ]
}
```

Jenkins Pipeline Configuration

Here's a sample pipeline configuration to get you started:

```
1    node {
2        stage('Cleanup') {
3            cleanWs()
4        }
5
6        stage('Git') {
7            dir('scm/my-git-repo') {
8                git(
9                    url: 'git@server.tld:account/my-git-repo.git',
10                   branch: 'develop',
11                   credentialsId: '22a0c94c-8f64-49eb-8385-bc99e08b3f10'
12               )
13           }
14       }
15
16       stage('Metadata') {
17           osfBuilderSuiteForSFCCDataImport(
18               hostname: 'cert.staging.realm.customer.demandware.net',
19               bmCredentialsId: '7fe08d04-87b1-49aa-9d8d-482bbc745faa',
20               tfCredentialsId: 'dc0a0c54-23b7-4879-a3e2-4f270eb668ca',
21               ocCredentialsId: 'a99c7226-6472-4051-8d7e-89af72f5daab',
22               ocVersion: 'v18_3',
23               archiveName: 'metadata',
24               sourcePath: 'scm/my-git-repo/metadata',
25               tempDirectory: 'tmp/data_import'
26           )
27       }
28   }
```

You can also always consult the pipelines documentation available at https://jenkins.io/doc/book/pipeline/ or check the pipeline syntax link right inside Jenkins on the left navigation menu.

Back to Dashboard

Status

Changes

Build Now

Delete Pipeline

Configure

Open Blue Ocean

Full Stage View

Pipeline Syntax

Git Polling Log

Version history

https://github.com/jenkinsci/osf-builder-suite-for-sfcc-data-import-plugin/releases