

Using Plugins in Shipbridge

04/30/2025 3:10 pm CDT

Overview

Shipbridge plugins can be used to apply shipping rules or customized invoices when shipping orders in Shipbridge. Shipbridge plugins are typically developed by Sellercloud and incur customization charges. Developers can use the instructions below to aid in development. Plugin files are with the extension .dll. Plugins are installed locally on your computer. If you have multiple computers set up with ShipBridge, you should install the plugin on each one of them.

Add Plugins in Shipbridge

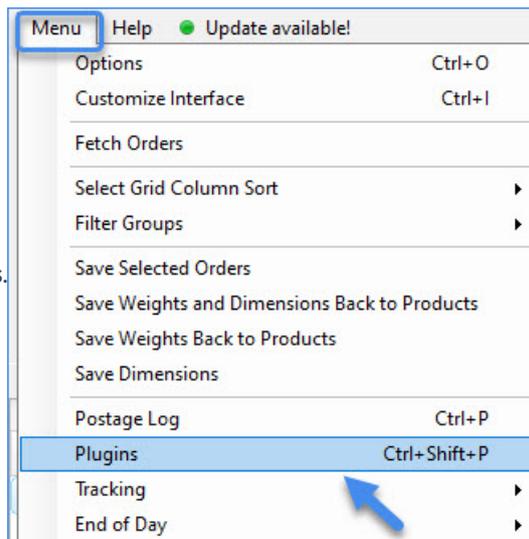
When you open your Shipbridge application, on the top-right corner you will see “No plugin enabled”. Please note that the text “No plugin installed” can be in red or black color. That varies depending is there an already available plugin that is not enabled or there is no plugin available in black color.

Please note that the plugin is available only on your local machine and exporting the Shipbridge settings file or your collected files will not carry the plugin along. If you are setting up a new Shipbridge on a new computer and you would like to have the same plugin active, please make sure to transfer the original plugin .dll file on the new machine and follow the above steps of installation.

To add plugins in Shipbridge:

1. Obtain your custom or generic plugin from Sellercloud Support by downloading the file from the support ticket or the email.
2. Save it to the PC where Shipbridge is installed.
3. In Shipbridge, open the **Plugins** window. This can be done either way:

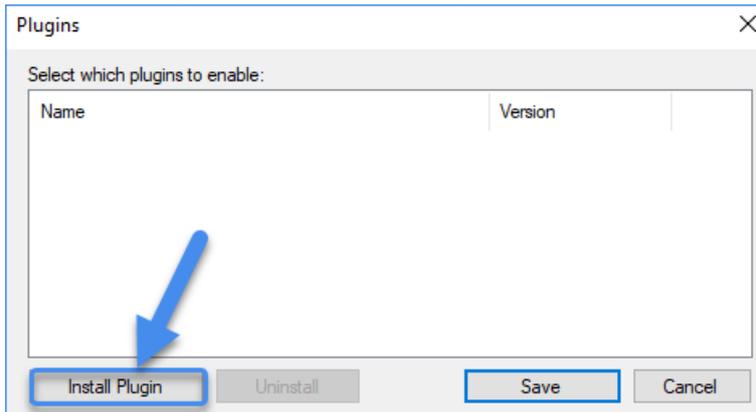
a. Go to **Menu** and select **Plugins**.



b. Use hotkey **Ctrl+Shift+P**.

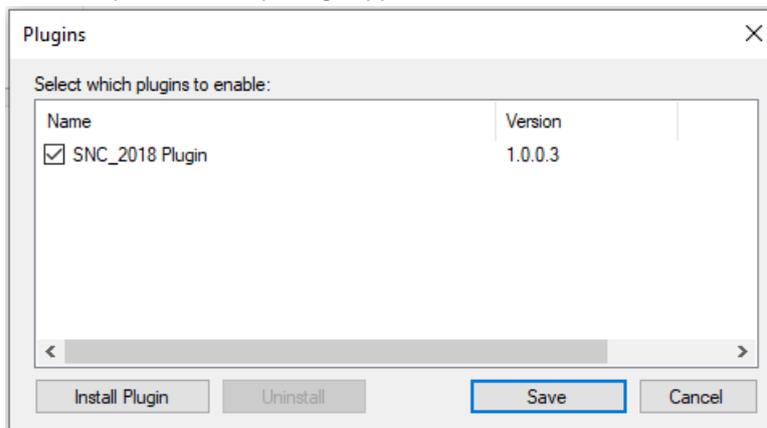
c. Click the **No plugin enabled** text on the top right.

4. Click **Install Plugin** and browse your computer for the .dll file.



5. Select the file and click open. That action will install the plugin onto your local Shipbridge application.

6. After installing the plugin, you need to enable the checkbox of the plugin in order to enable it and make it active on your local Shipbridge application.



Please note that you can enable only one plugin at a time.

Shipbridge will call all enabled plugins to retrieve a shipping rule for each order. The plugins will be called whenever a new order is loaded to, or refreshed in the grid, or scanned in Scan & Ship.

Write Plugins

The **Shipbridge Extensibility Framework** provides the `IExtensibility` interface— a proxy between plugins and Shipbridge. Extensibility allows quick access to such features as the “grids” and other shared instances. Plugins are essentially .NET 2.0 assemblies. See attached sample VB.NET plugin below, `Shipbridge.Plugin.Sample`, which you can customize.

Extensibility in Shipbridge

The most important reference is `Shipbridge.Extensibility.dll` is found in the Shipbridge application folder:

- It exposes the `IPlugin` interface which all plugins should implement.
- It provides access to `ShipperHelper` and `PluginHelper` which have easy-to-use properties and methods for checking orders.

Plugin requirements

Plugin requirements:

1. ID – Each plugin should have a unique identifier (GUID). You can generate one from [here](#).
2. DisplayName and Version keep the Plugins window updated with relevant information
3. Handlers:
 - a. OnConnected is called every time Shipbridge initializes the plugin.
 - b. OnOrderBeforeAddedToGrid is called immediately after a new order is received, but before any processing is made by Shipbridge itself. This is useful if a shipping method has to be selected before Shipbridge processes the order.
 - c. OnOrderAfterAddedToGrid is the most frequently used handler. It's called after Shipbridge has finished processing the order and allows plugins to select a proper shipping method.
 - d. OnCellBeginEdit is called when the user starts making a change to a grid cell. This handler is used rarely.
 - e. OnCellEndEdit is called after the user makes a change to a grid cell. This handler allows you to handle any address or weight changes so that the proper shipping method is selected.

PluginHelper

PluginHelper provides quick access to frequently used order operations, getting or checking the state against another value, calculating its subtotal, weight, or checking if the order is “international,” to a military/US territory address, and so on.

Instantiating a PluginHelper is done with a simple call by referencing the **OrderData**: Dim helper As New PluginHelper(data).

The actual change of shipping service can be made by changing the helper. ShippingService to a ServiceInfo instance using ServiceInfo.GetByName. You can find a full ServiceInfo listing below. Of course, you can use any shipping rules based on the destination, order subtotal, weight, etc. [Click here for an example.](#)

Notes

It's important to skip fully shipped or “ready for pickup” orders so that their shipping service is not altered. This can be done using the following code as used in the sample plugin.

- `' SkipFullyShipped and ReadyForPickup orders If helper.ShippingStatus = OrderShippingStatus.FullyShipped OrElse helper.ShippingStatus = OrderShippingStatus.ReadyForPickup Then Return False End If`

You can optionally skip rush or other orders by checking with the helper or another OrderData property.

- `' Leave this uncommented if rush orders should be skipped If helper.RushOrder Then Return False End If`

ServiceInfo

You can get a ServiceInfo instance of any shipping service by calling the ServiceInfo.GetByName method (case-insensitive).

[Click here for a list of carrier names.](#)

UPS Expedited Mail Innovations is the only MI method available to domestic (US) destinations. Similarly, UPS Economy Mail Innovations is only available to international destinations.

Attachments

[ShipBridge Plugin Sample](#)
