

Table of Contents

| | |
|---|----|
| Integrating the cartridge | 2 |
| Assigning the cartridges to your sites..... | 2 |
| Setting up the Footprints Service | 3 |
| Integrating the styling and javascript..... | 5 |
| Adding template changes..... | 6 |
| Integrating with Controllers..... | 7 |
| Integrating with Pipelines | 8 |
| File Management..... | 10 |

INSTALLATION GUIDE

Integrating the cartridge

This guide contains the details on how to integrate the Footprints cartridge to your code base.

Assigning the cartridges to your sites

The cartridge uses the Service Framework by Salesforce Commerce Cloud.

Complete the following steps to assign the cartridge to your site and Business Manager:

1. Obtain the `footprints.zip` file.
2. Extract archive to your local file system – e.g. the cartridge folder of your project.
3. Import cartridges into your workspace and link them to the Digital Server Connection.
4. Assign the cartridges to your site's cartridge path. Add below cartridges to the effective cartridge path.

```
int_footprints_pipelines:int_footprints for projects using Pipelines  
int_footprints_controllers:int_footprints for projects using Controllers.
```

Setting up the Footprints Service

1. Navigate to BM > Administration > Operations > Services.
2. Go to the Credentials tab and create a new Service Credentials entry called FootprintsCredentials with the below data

| Attribute | Value |
|-----------|---|
| Name | FootprintsCredentials |
| URL | https://testapi.footprintsnetwork.org/v1/ |
| User | Username received from Footprints |
| Password | Password |

3. Go back to the Services page and go to the Profiles tab and create a new Service Profile entry called FootprintsProfile with the below data:

| Attribute | Value |
|-------------------------------|-------------------|
| Name | FootprintsProfile |
| Timeout(ms) | 30000 |
| Enable Circuit Breaker | Checked |
| Max Circuit Breaker Calls | 3 |
| Circuit Breaker Interval (ms) | 10000 |
| Enable Rate Limit | |
| Max Rate Limit Calls | |
| Rate Limit Interval (ms) | |

4. Go back to the Services page and go to the Services tab and create a new Services entry called FootprintsService with the below data:

| Attribute | Value |
|-----------------|-------------------|
| Name | FootprintsService |
| Type | HTTP |
| Enabled | Checked |
| Service Mode | Live |
| Log Name Prefix | FootprintsService |

| | |
|---------------------------|-----------------------|
| Communication Log Enabled | Checked |
| Profile | FootprintsProfile |
| Credentials | FootprintsCredentials |

Integrating the styling and javascript

1. Using grunt/gulp

For using the styling, copy the css from footprints.css into your `_checkout.scss` file. The css written in footprints.css is a valid scss.

For using the javascript, you can call the footprints.js in the checkout footer of your application.

2. Using standard js/css

For using the styling, either copy the css from footprints.css into your css at the bottom or reference the footprints.css in the checkout header of your application.

For using the javascript, call the footprints.js in the checkout footer of your application.

NOTE: the javascript and styling is written specifically for the html elements being used to show the Footprints data on the billing page. This should not interfere with your application javascript or styling.

Adding template changes

You'll need to add the below mentioned changes in the templates:

1. Displaying the donation projects on the billing page

Put the below code in the **billing.isml** before the html content which displays the coupon codes

```
<iscomment>integrating_footprints</iscomment>
<isinclude template="billing/footprintsDonations" />
```

2. Displaying the chosen donation on the order summary

Put the below code in the ordertotals.isml before the html content which displays the taxation.

```
<iscomment>integrating_footprints</iscomment>
<isscript>
  var allPAs = LineItemCtrn.getPriceAdjustments();
  var microdonations = 0;
  for each(var pa in allPAs ){
    if(pa.getPromotionID().indexOf('footprints') != -1) {
      microdonations = microdonations + pa.getPriceValue();
    }
  }
</isscript>
<isif condition="${microdonations > 0}">
  <tr class="microdonations">
    <td>Micro-donations</td>
    <td>
      <isprint value="${new dw.value.Money(microdonations, session.currency.currencyCode)}"/>
    </td>
  </tr>
</isif>
```

3. Displaying the chosen donation in the order confirmation email

Put the below code in the ordertotalsemail.isml before the html content which displays the taxation.

```
<iscomment>integrating_footprints</iscomment>
<isscript>
  var allPAs = LineItemCtrn.getPriceAdjustments();
  var microdonations = 0;
  for each(var pa in allPAs ){
    if(pa.getPromotionID().indexOf('footprints') != -1) {
      microdonations = microdonations + pa.getPriceValue();
    }
  }
</isscript>
<isif condition="${microdonations > 0}">
  <tr class="microdonations">
    <td>Micro-donations</td>
    <td>
      <isprint value="${new dw.value.Money(microdonations, session.currency.currencyCode)}"/>
    </td>
  </tr>
</isif>
```

Integrating with Controllers

You need to integrate the cartridge for two things:

1. Getting the list of supported donations on the billing page

Put the below code after the `/* script includes */` section in your **COBilling.js** controller.

```
// integrating_footprints
var Status = require('dw/system/Status');
var FPHelper = require("int_footprints/cartridge/scripts/utils/FPHelper");
var fpHelper = new FPHelper();
```

Put the below code in the `publicStart()` function just before the code calls `start()` function.

```
// integrating_footprints
fpHelper.setItemsForCurrentSession();

start(cart, {ApplicableCreditCards: creditCardList.ApplicableCreditCards});
```

2. Sending the saved donations in the basket to the Footprints.

Put the below code after `/* script includes */` section in your **COPlaceOrder.js** controller.

```
// integrating_footprints
importScript('int_footprints:utils/FPUtils.ds');
var Status = require('dw/system/Status');
var FPHelper = require("int_footprints/cartridge/scripts/utils/FPHelper");
var fpHelper = new FPHelper();
```

Put the below code in the `start()` function above the return statement.

```
// integrating_footprints
var postData = generatePostData(order);
fpHelper.postOrderData(postData);

return orderPlacementStatus;
```

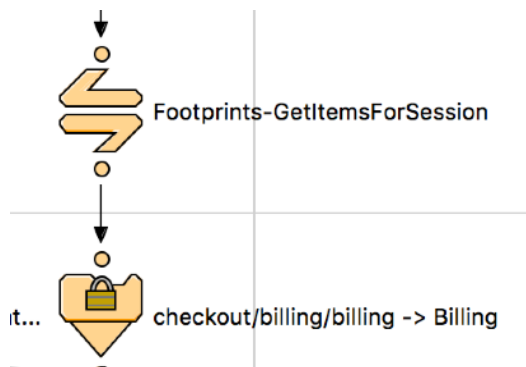
NOTE: There are two sample files in the `int_footprints_controllers` cartridge for reference. `COBilling.js` & `COPlaceOrder.js`.
Search for `// integrating_footprints`

Integrating with Pipelines

You need to integrate the cartridge for two things:

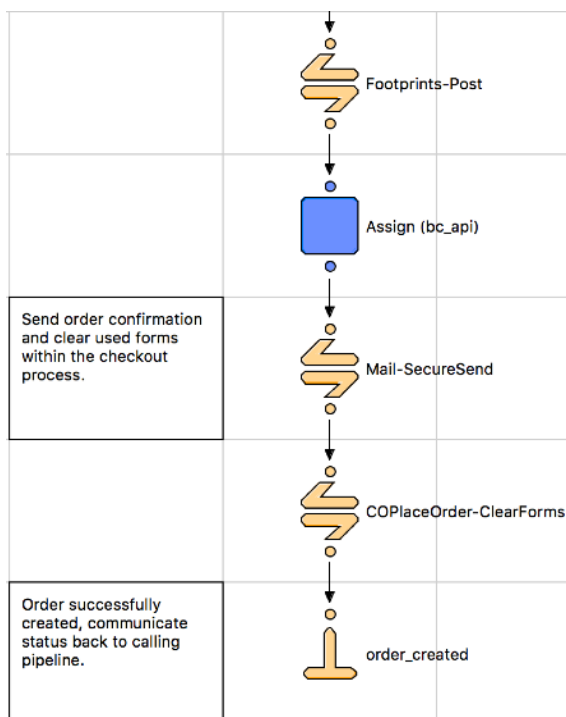
1. Getting the list of supported donations on the billing page

Add a call node to the Footprints-GetItemsForSession pipeline just before rendering the billing.isml in COBilling-Start pipeline.



2. Sending the saved donations in the basket to the Footprints.

Add a call node to Footprints-Post pipeline after before sending the confirmation email in the COPlaceOrder-Start pipeline.



NOTE: There are two sample files in the int_footprints_pipelines cartridge for reference. COBilling.xml & COPlaceOrder.xml.

File Management

The cartridge doesn't create any files in any of the WEBDAV folder on the Salesforce Commerce Cloud instances.