SFRA (Storefront Reference Architecture) Version

# PEPPERJAM-SALESFORCE COMMERCE CLOUD INTEGRATION

09.07.18

Presented by

pepperjam®

# Contents

## Summary

This document provides technical instructions for using the Pepperjam Salesforce Commerce Cloud Cartridge to integrate the Pepperjam Dynamic Advertiser Pixel, Gateway Advertiser and Pixel Tag Container into a Salesforce Commerce Cloud storefront. It also provides access to Advanced Product Feed and Sales Correction Feed for the Pepperjam service.

## Glossary

| Term | Description |
|---|---|
| SFCC | Salesforce Commerce Cloud (formerly Demandware) |
| Business Manager (BM) | The primary tool used to configure the SFCC platform and customer storefront |
| SFRA | Storefront Reference Architecture |

## Version

| Version | Last Updated | Notes |
|---|---|---|
| V1.0 | 8/13/2018 | Initial version |
| V1.1 | 9/7/2018 | Updated to accommodate Apple ITP |

# 1 Component Overview

### 1.1 FUNCTIONAL OVERVIEW

The Pepperjam cartridge enables a merchant to set the Pepperjam Dynamic Advertiser Pixel, Gateway Advertiser and Pixel Tag Container in SFCC and manage each by selecting enable or disable.   The cartridge also generates a product feed.  Additionally an optional transaction error and correction feed is generated that shows orders with corrections (like returns).

### 1.2 COMPATABILITY

The cartridge is designed for Salesforce Commerce Cloud API version 18.6 with controllers. Compatibility Mode: 16.2. It is expected to be backward compatible with older versions of SiteGenesis. The cartridge utilizes the Pepperjam Dynamic Pixel.

### 1.3 PRIVACY, PAYMENT

Pepperjam's cartridge does not collect and process any user profile information or billing information.

## 2 Pre-work

Use of the Pepperjam SFCC Cartridge requires credentials from Pepperjam. Please contact your Pepperjam account team or launch team representative to receive:

- Pepperjam Program ID
- Pepperjam Tag ID

If you are contacting Pepperjam for the first time, please reach out to sfccpartners@pepperjam.com.

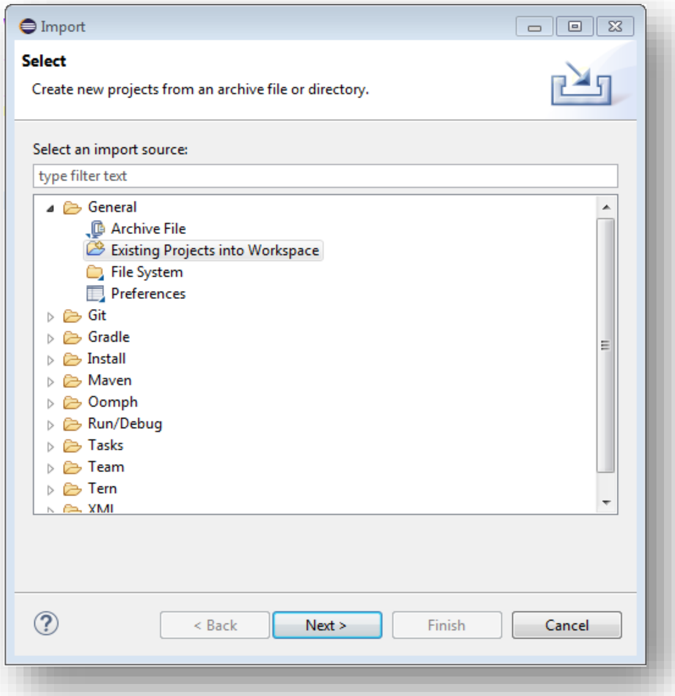
From your internal team:

- Optional: If you are setting up the order correction feed, you'll need an FTP site that Pepperjam can pick the feed up from. Therefore, gather these FTP credentials prior to starting.

# 3 Implementation Guide

## 3.1 GENERAL IMPLEMENTATION: INSTALLING THE CARTRIDGE ON THE SANDBOX

1. Open Demandware Studio.
2. Import the downloaded cartridge. On the import screen, select: Existing Projects into Workspace



Import

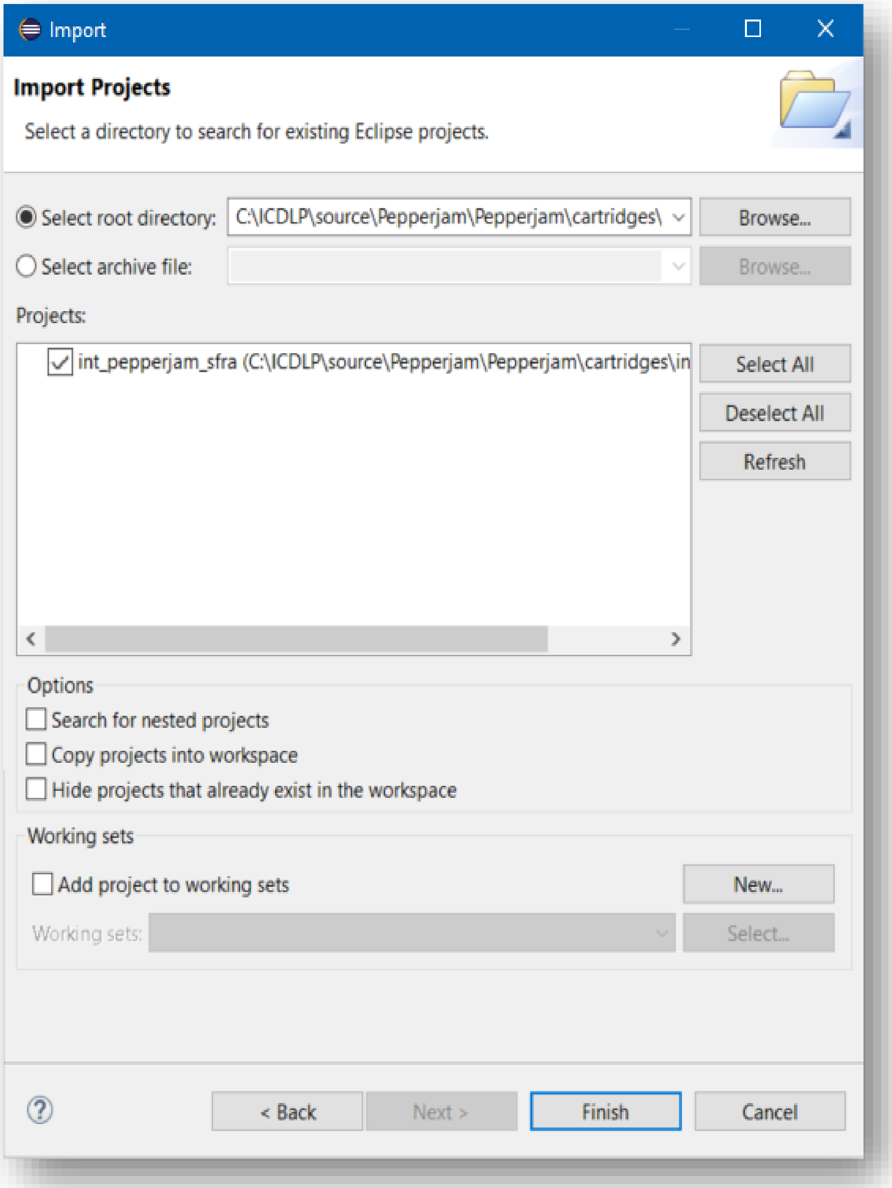
Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

3. Link the cartridge to the sandbox



Import

**Import Projects**

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\ICDLP\source\Pepperjam\Pepperjam\cartridges\

○ Select archive file:

Projects:

☑ int_pepperjam_sfra (C:\ICDLP\source\Pepperjam\Pepperjam\cartridges\in

Options

□ Search for nested projects

□ Copy projects into workspace

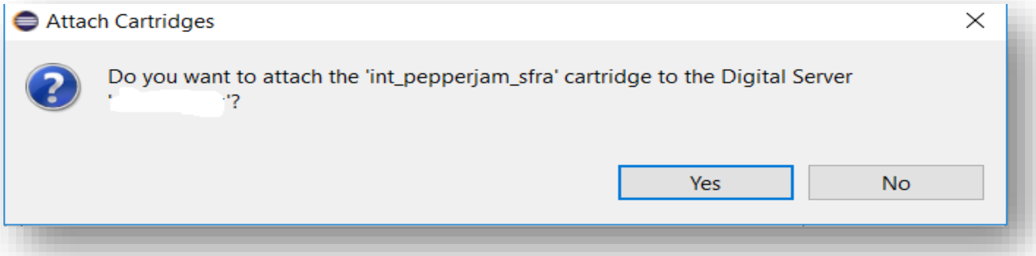
□ Hide projects that already exist in the workspace

Working sets

□ Add project to working sets

Working sets:

4. Select sandbox connection, then Properties. Select Project Reference and check in.



Attach Cartridges

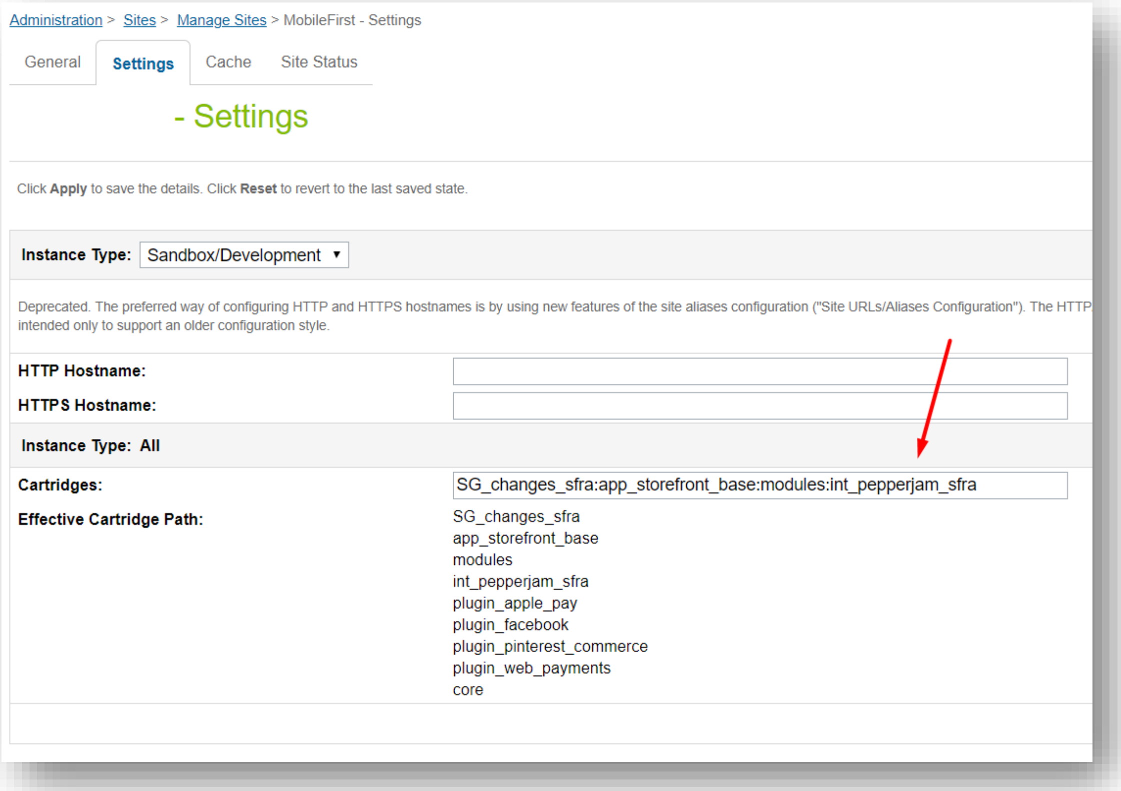
Do you want to attach the 'int_pepperjam_sfra' cartridge to the Digital Server '[redacted]'?

## 3.2 SET UP BUSINESS MANAGER

### 3.2.1 Manage Sites Settings

Go to Business Manager -> Site -> Manage Sites. Select the correct site, then select **Settings** tab. In the cartridge path add the following line:

SG_changes_sfra:app_storefront_base:modules:int_pepperjam_sfra



Administration > Sites > Manage Sites > MobileFirst - Settings
General
Settings
Cache
Site Status

- Settings

Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

**Instance Type:** Sandbox/Development ▾

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP intended only to support an older configuration style.

**HTTP Hostname:**

**HTTPS Hostname:**

**Instance Type: All**

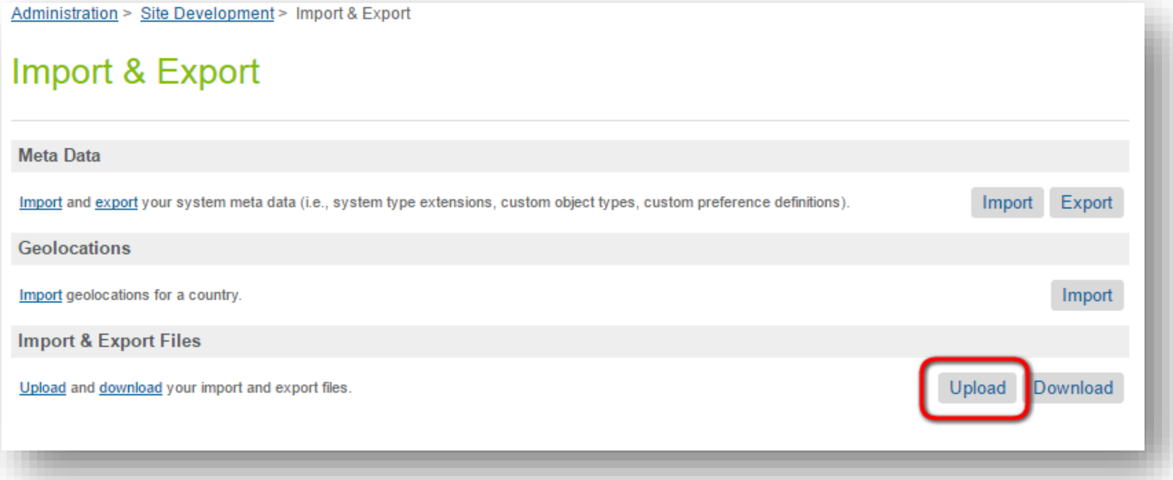
**Cartridges:** SG_changes_sfra:app_storefront_base:modules:int_pepperjam_sfra

**Effective Cartridge Path:**
SG_changes_sfra
app_storefront_base
modules
int_pepperjam_sfra
plugin_apple_pay
plugin_facebook
plugin_pinterest_commerce
plugin_web_payments
core

## 3.2.2 Import and Export Settings

### *3.2.2.1 System Object Type Extensions*

Go to Administration -> Site Development -> Import & Export. Upload system-objecttype-extensions.xml from the metadata folder.



Administration > Site Development > Import & Export

Import & Export

Meta Data

Import and export your system meta data (i.e., system type extensions, custom object types, custom preference definitions).

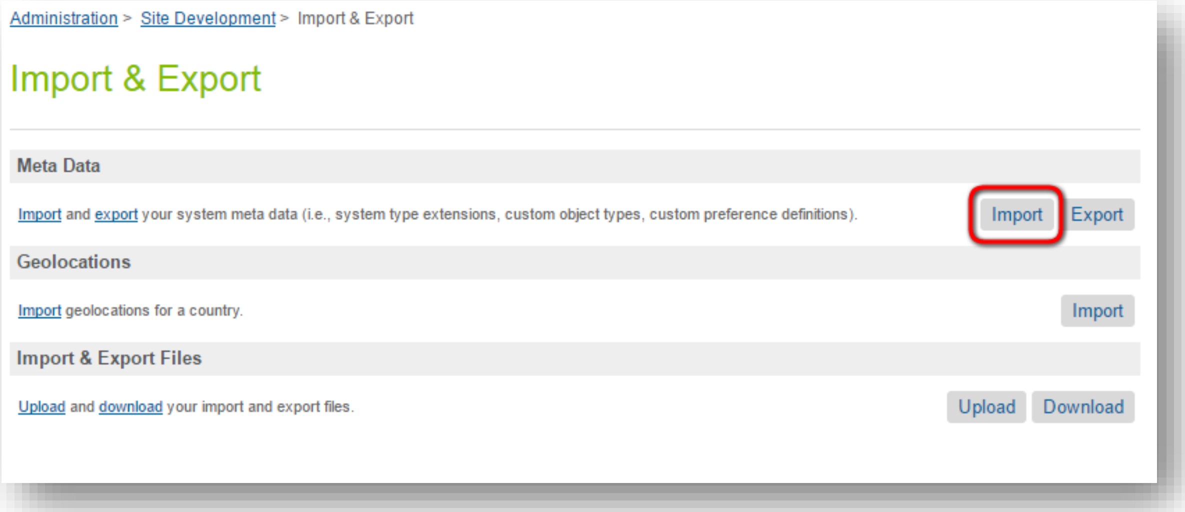
Geolocations

Import geolocations for a country.

Import & Export Files

Upload and download your import and export files.

Click on the import button to follow steps to import system-objecttype-extensions.xml



Administration > Site Development > Import & Export

Import & Export

Meta Data

Import and export your system meta data (i.e., system type extensions, custom object types, custom preference definitions).

Geolocations

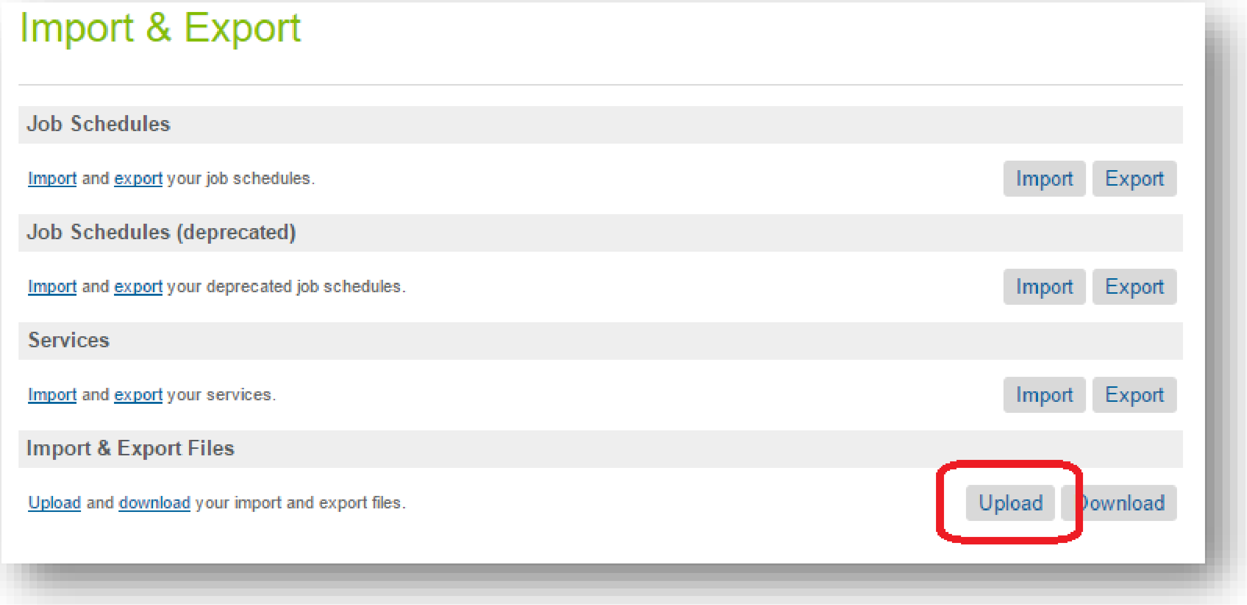
Import geolocations for a country.

Import & Export Files

Upload and download your import and export files.

### *3.2.2.2 Services*

Go to Administration > Operations > Import & Export and upload services.xml from the metadata folder.



Import & Export

Job Schedules

Import and export your job schedules.

Job Schedules (deprecated)

Import and export your deprecated job schedules.

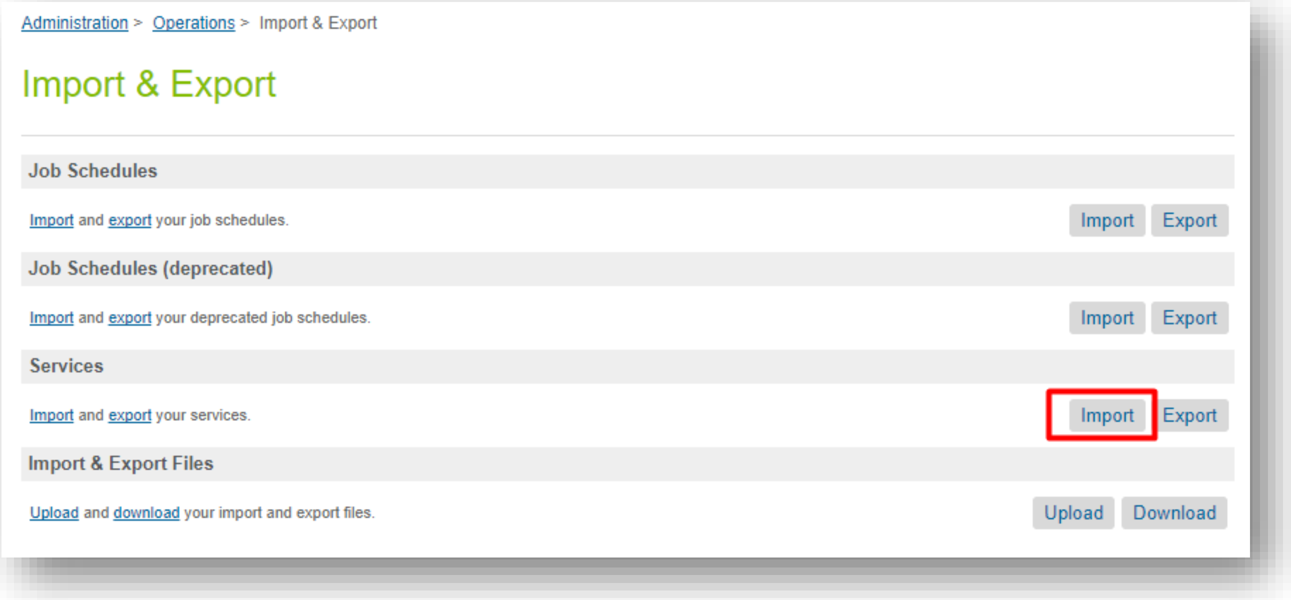
Services

Import and export your services.

Import & Export Files

Upload and download your import and export files.

Click on Import and follow steps to import services.xml.



Administration > Operations > Import & Export

Import & Export

Job Schedules

Import and export your job schedules.

Job Schedules (deprecated)

Import and export your deprecated job schedules.

Services

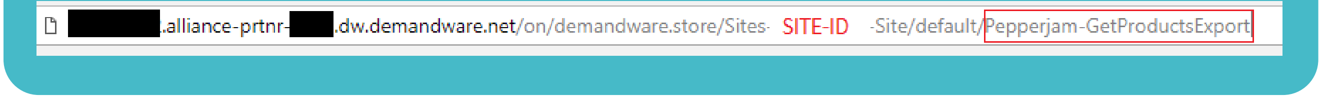
Import and export your services.

Import & Export Files

Upload and download your import and export files.

### *3.2.2.3 Configure Jobs.xml for Product Feed and Order Correction Feed*

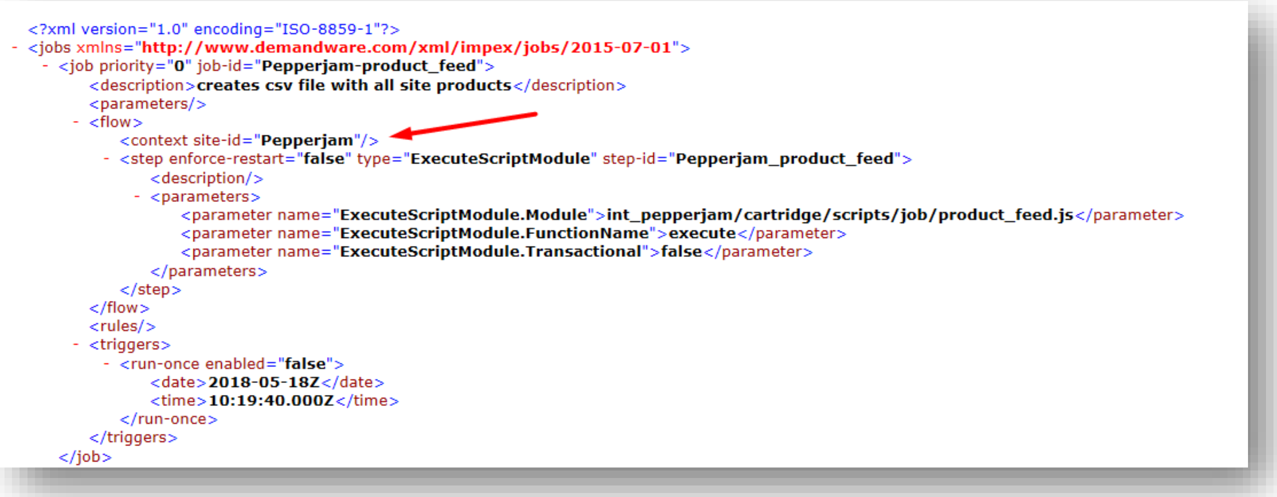
Reference: To get the link for the HTTP GET access to the product feed, add “Pepperjam-GetProductsExport” to your main site address as shown below



```
[redacted].alliance-prtnr-[redacted].dw.demandware.net/on/demandware.store/Sites- SITE-ID -Site/default/Pepperjam-GetProductsExport
```

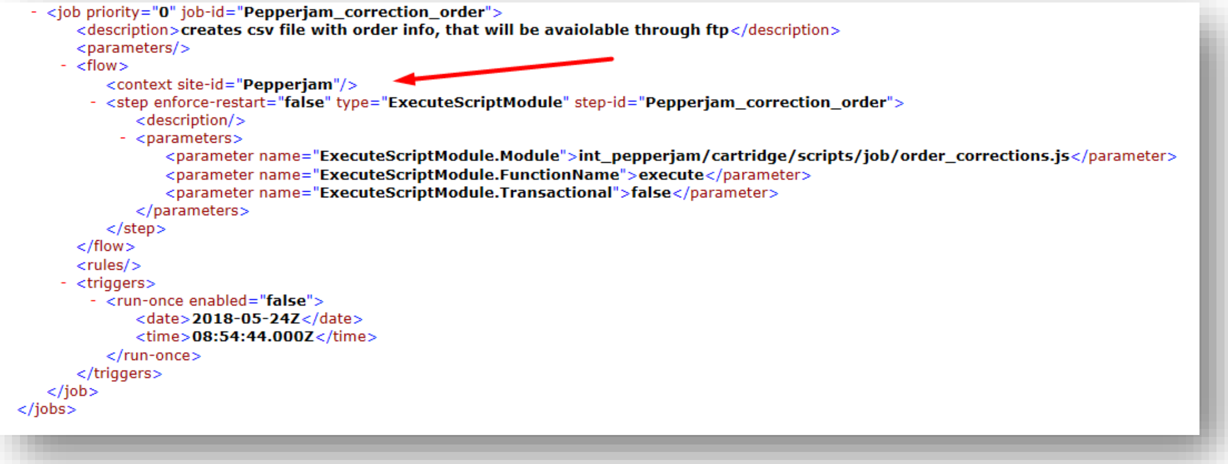
In the metadata folder in the file jobs.xml you need to define your siteID for the Product Feed and Order Corrections Feed jobs as shown below (NOTE: The Order Corrections Feed is Optional).

Product Feed:



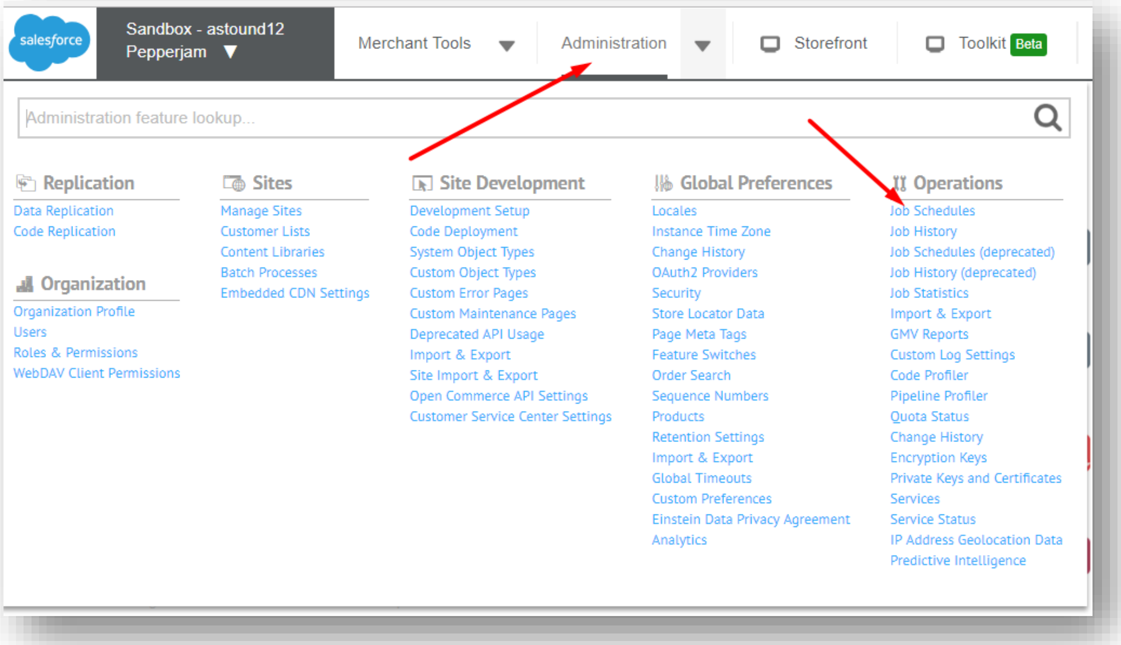
```
   <?xml version="1.0" encoding="ISO-8859-1"?>
 - <jobs xmlns="http://www.demandware.com/xml/impex/jobs/2015-07-01">
   - <job priority="0" job-id="Pepperjam-product_feed">
       <description>creates csv file with all site products</description>
       <parameters/>
     - <flow>
         <context site-id="Pepperjam"/>
       - <step enforce-restart="false" type="ExecuteScriptModule" step-id="Pepperjam_product_feed">
           <description/>
         - <parameters>
             <parameter name="ExecuteScriptModule.Module">int_pepperjam/cartridge/scripts/job/product_feed.js</parameter>
             <parameter name="ExecuteScriptModule.FunctionName">execute</parameter>
             <parameter name="ExecuteScriptModule.Transactional">false</parameter>
           </parameters>
         </step>
       </flow>
       <rules/>
     - <triggers>
       - <run-once enabled="false">
           <date>2018-05-18Z</date>
           <time>10:19:40.000Z</time>
         </run-once>
       </triggers>
     </job>
```

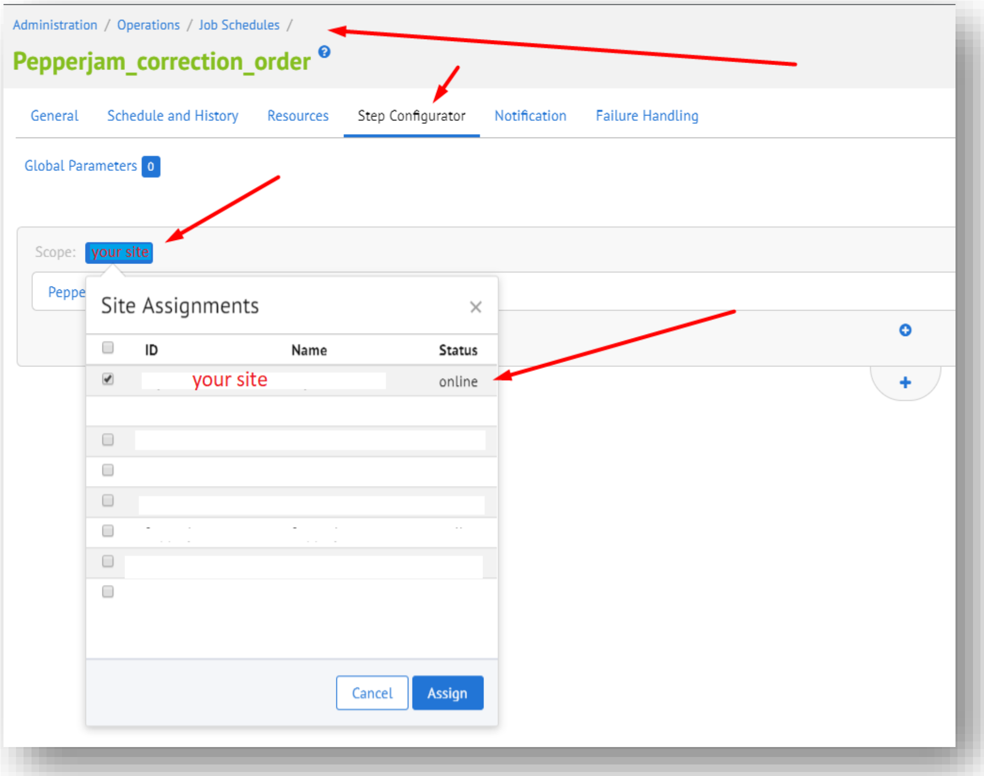
Order Feed (Optional – see section 3.2.3):



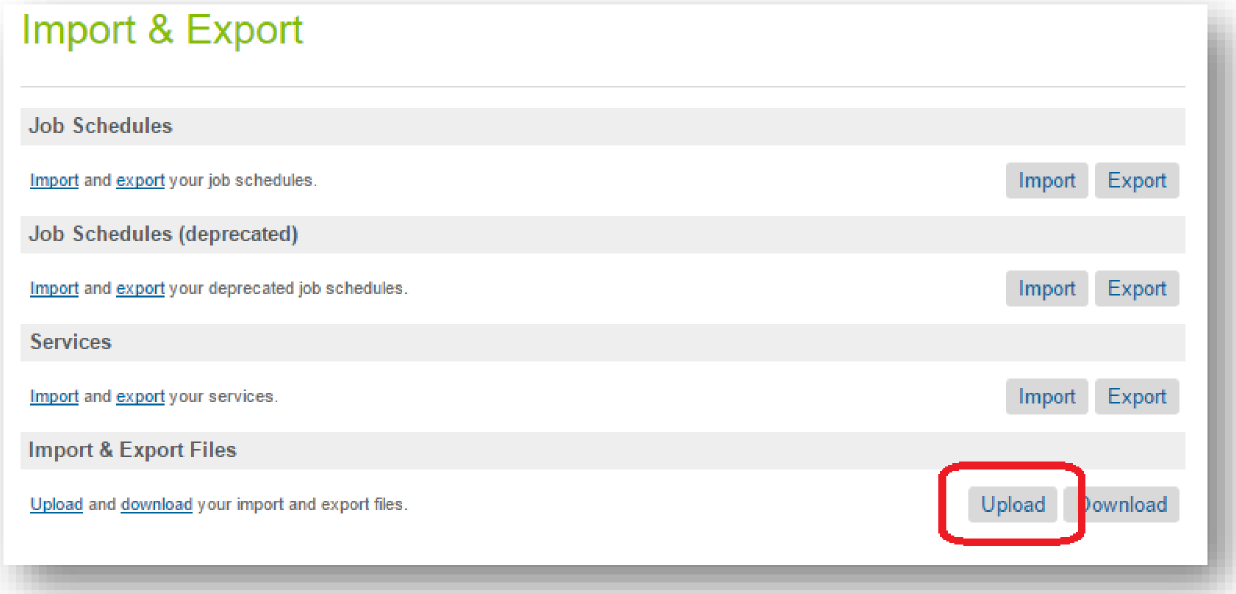
```
  - <job priority="0" job-id="Pepperjam_correction_order">
      <description>creates csv file with order info, that will be avaiolable through ftp</description>
      <parameters/>
    - <flow>
        <context site-id="Pepperjam"/>
      - <step enforce-restart="false" type="ExecuteScriptModule" step-id="Pepperjam_correction_order">
          <description/>
        - <parameters>
            <parameter name="ExecuteScriptModule.Module">int_pepperjam/cartridge/scripts/job/order_corrections.js</parameter>
            <parameter name="ExecuteScriptModule.FunctionName">execute</parameter>
            <parameter name="ExecuteScriptModule.Transactional">false</parameter>
          </parameters>
        </step>
      </flow>
      <rules/>
    - <triggers>
      - <run-once enabled="false">
          <date>2018-05-24Z</date>
          <time>08:54:44.000Z</time>
        </run-once>
      </triggers>
    </job>
</jobs>
```

Or you can manually configure these settings in the Job Schedules section within BM after the import of jobs.xml as shown below:





Go to Administration > Operations > Import & Export and upload jobs.xml from the metadata folder.



Import & Export

Job Schedules

Import and export your job schedules.

Job Schedules (deprecated)

Import and export your deprecated job schedules.

Services

Import and export your services.

Import & Export Files

Upload and download your import and export files.

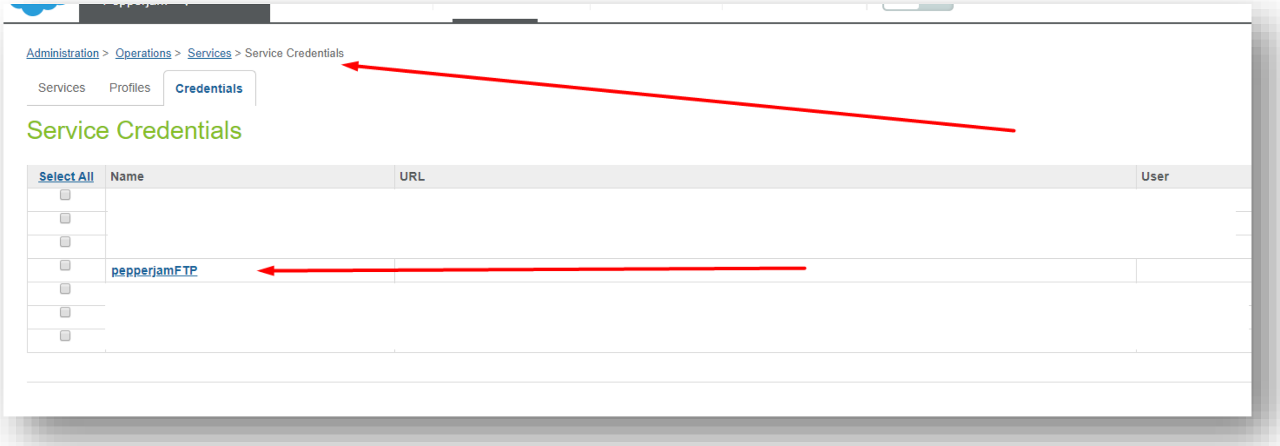
Click on Import and follow the steps to import jobs.xml.

### 3.2.3 Configure Order Correction Feed (OPTIONAL)

Pepperjam can receive order corrections such as full returns/cancellations to automatic reflect in the network within the transaction locking period to ensure advertisers do not pay commissions on transactions that customers returned or were cancelled. Pepperjam's Salesforce Commerce Cloud cartridge is able to process full order corrections, but not partial order corrections.

NOTE: If you use Salesforce Commerce Cloud's Order Management System (called SFCC OMS), then follow the steps below to configure the order correction feed for Pepperjam. For other order management systems, contact your Pepperjam account strategist or Pepperjam launch team representative for more information.

Before beginning this section, make sure you have your FTP credentials ready. In Business Manager go to Administration —> Operations —>Services then choose tab credentials:

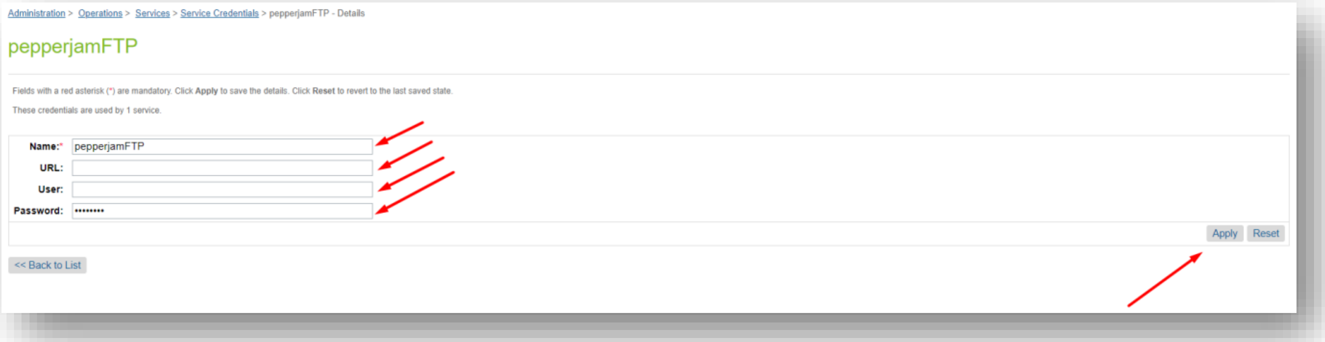


Administration > Operations > Services > Service Credentials

Service Credentials

| Select All | Name | URL | User |
|---|---|---|---|
| □ | | | |
| □ | | | |
| □ | | | |
| □ | pepperjamFTP | | |
| □ | | | |
| □ | | | |
| □ | | | |

Choose ‘pepperjamFTP’ and enter your credentials.  Press save.



Administration > Operations > Services > Service Credentials > pepperjamFTP - Details

pepperjamFTP

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

These credentials are used by 1 service.

| | |
|---|---|
| **Name:*** | pepperjamFTP |
| **URL:** | |
| **User:** | |
| **Password:** | •••••••• |

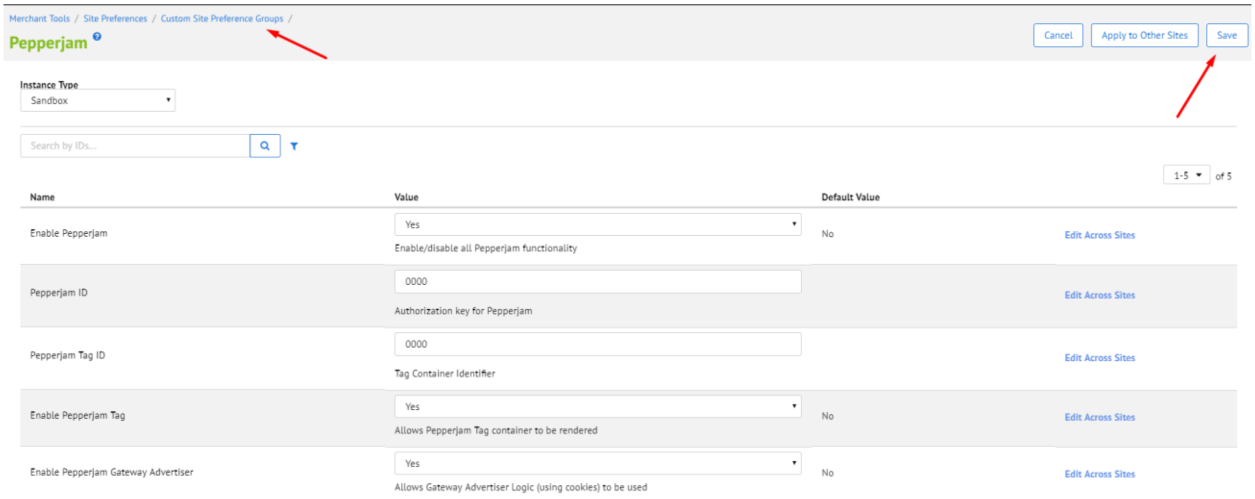
### 3.2.4 Configure Custom Site Preferences for Pepperjam

Go to Business Manager: Sites -> Your Site -> Site Preferences -> Custom Site Preferences -> Pepperjam.

| Name | Description |
|---|---|
| Enable Pepperjam | Turns on\off the Pepperjam integration. Set this value to Yes. |
| Pepperjam ID | Pepperjam Program ID (PID). You can get this from your Pepperjam account representative or Pepperjam launch team member. |
| Pepperjam Tag ID | Tag ID for the Pepperjam Pixel Tag Container which is provided by Pepperjam. You can get this from your Pepperjam account representative or Pepperjam launch |

| | |
|---|---|
| | team member. |
| Enable Pepperjam Tag | Turns on\off the Pepperjam Pixel Tag Container. This should always be set to Yes |
| Enable Pepperjam Gateway Advertiser | Turns on\off the Pepperjam Gateway Advertiser. This should always be set to Yes. |

Below is an example of how to complete the fields in the administration area. Press save when you are done.



Pepperjam

Instance Type: Sandbox

| Name | Value | Default Value |
|---|---|---|
| Enable Pepperjam | Yes<br>Enable/disable all Pepperjam functionality | No |
| Pepperjam ID | 0000<br>Authorization key for Pepperjam | |
| Pepperjam Tag ID | 0000<br>Tag Container Identifier | |
| Enable Pepperjam Tag | Yes<br>Allows Pepperjam Tag container to be rendered | No |
| Enable Pepperjam Gateway Advertiser | Yes<br>Allows Gateway Advertiser Logic (using cookies) to be used | No |

## 3.3 INTEGRATE PEPPERJAM SETTINGS INTO YOUR STOREFRONT

To integrate Pepperjam into your storefront you need to work with a developer on your team to insert code into specific templates in the core cartridge for:

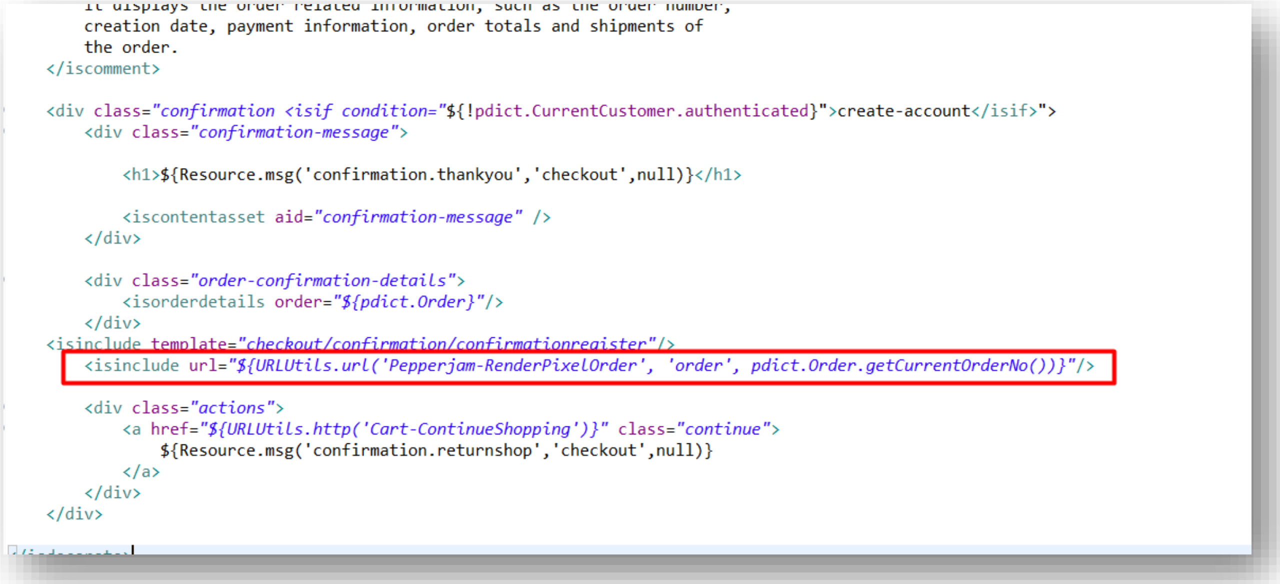
- Pepperjam Dynamic Advertiser Pixel
- Pepperjam Gateway Advertiser Pixel
- Pepperjam Tag Container

### 3.3.1 Pepperjam Dynamic Advertiser Pixel

For the Dynamic Advertiser Pixel, insert the code below into the

checkout/confirmation/confirmation.isml template

```
<isinclude url="${URLUtils.url('Pepperjam-RenderPixelOrder', 'order',
pdict.Order.getCurrentOrderNo())}"/>
```

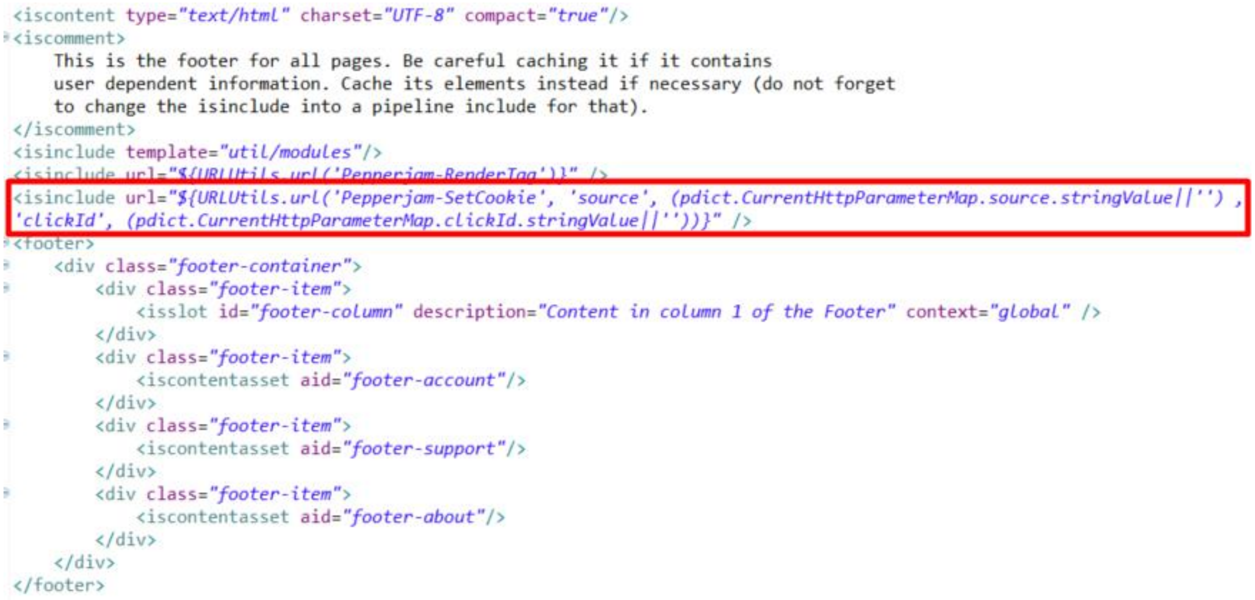


```
        It displays the order related information, such as the order number,
        creation date, payment information, order totals and shipments of
        the order.
    </iscomment>

    <div class="confirmation <isif condition="${!pdict.CurrentCustomer.authenticated}">create-account</isif>">
        <div class="confirmation-message">

            <h1>${Resource.msg('confirmation.thankyou','checkout',null)}</h1>

            <iscontentasset aid="confirmation-message" />
        </div>

        <div class="order-confirmation-details">
            <isorderdetails order="${pdict.Order}"/>
        </div>
    <isinclude template="checkout/confirmation/confirmationregister"/>
        <isinclude url="${URLUtils.url('Pepperjam-RenderPixelOrder', 'order', pdict.Order.getCurrentOrderNo())}"/>

        <div class="actions">
            <a href="${URLUtils.http('Cart-ContinueShopping')}" class="continue">
                ${Resource.msg('confirmation.returnshop','checkout',null)}
            </a>
        </div>
    </div>

</isdecorate>
```

### 3.3.2 Pepperjam Gateway Advertiser Pixel

For the Gateway Advertiser Pixel, insert the code below into the components/footer/footer.isml template

*<isinclude url="${URLUtils.url('Pepperjam-SetCookie', 'source', (pdict.CurrentHttpParameterMap.source.stringValue||") , 'clickId', (pdict.CurrentHttpParameterMap.clickId.stringValue||"))}" />*

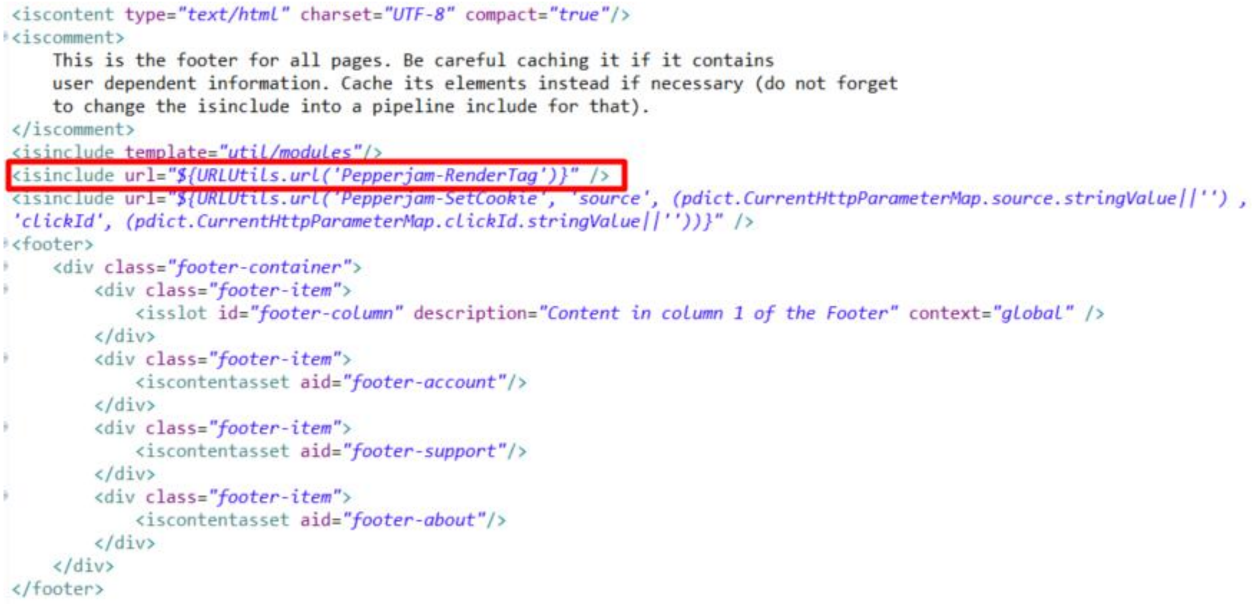


```
<iscontent type="text/html" charset="UTF-8" compact="true"/>
<iscomment>
    This is the footer for all pages. Be careful caching it if it contains
    user dependent information. Cache its elements instead if necessary (do not forget
    to change the isinclude into a pipeline include for that).
</iscomment>
<isinclude template="util/modules"/>
<isinclude url="${URLUtils.url('Pepperjam-RenderTag')}" />
<isinclude url="${URLUtils.url('Pepperjam-SetCookie', 'source', (pdict.CurrentHttpParameterMap.source.stringValue||'') ,
'clickId', (pdict.CurrentHttpParameterMap.clickId.stringValue||''))}" />
<footer>
    <div class="footer-container">
        <div class="footer-item">
            <isslot id="footer-column" description="Content in column 1 of the Footer" context="global" />
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-account"/>
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-support"/>
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-about"/>
        </div>
    </div>
</footer>
```

### 3.3.3 Pepperjam Tag Container

For the Tag Container, insert the code below into the components/footer/footer.isml template above the Gateway Advertiser pixel

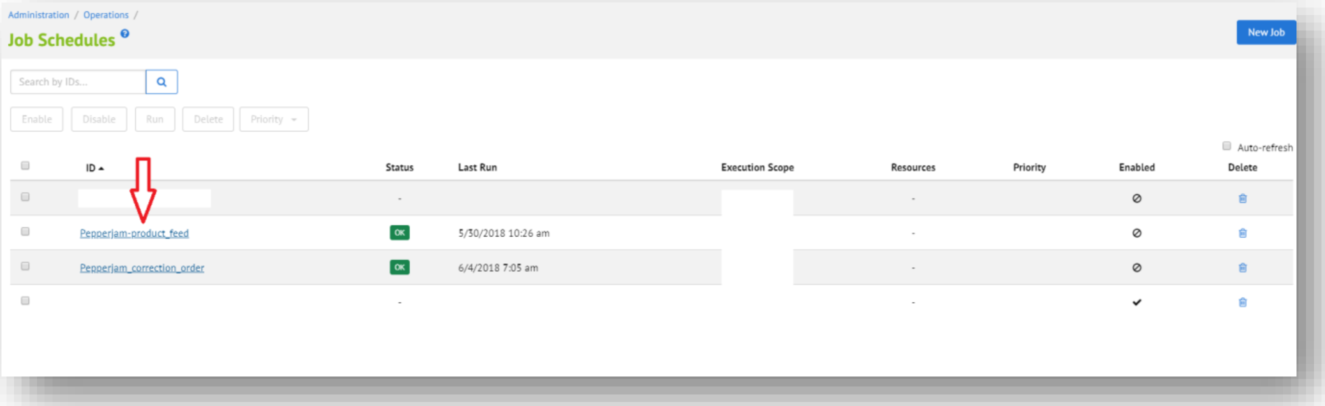
**<isinclude url="${URLUtils.url('Pepperjam-RenderTag')}" />**



```
<iscontent type="text/html" charset="UTF-8" compact="true"/>
<iscomment>
    This is the footer for all pages. Be careful caching it if it contains
    user dependent information. Cache its elements instead if necessary (do not forget
    to change the isinclude into a pipeline include for that).
</iscomment>
<isinclude template="util/modules"/>
<isinclude url="${URLUtils.url('Pepperjam-RenderTag')}" />
<isinclude url="${URLUtils.url('Pepperjam-SetCookie', 'source', (pdict.CurrentHttpParameterMap.source.stringValue||'') ,
'clickId', (pdict.CurrentHttpParameterMap.clickId.stringValue||''))}" />
<footer>
    <div class="footer-container">
        <div class="footer-item">
            <isslot id="footer-column" description="Content in column 1 of the Footer" context="global" />
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-account"/>
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-support"/>
        </div>
        <div class="footer-item">
            <iscontentasset aid="footer-about"/>
        </div>
    </div>
</footer>
```

# 4 Reference: Feed Jobs Run

After all imports of metadata are complete and credentials are configured in site preferences, you can view the jobs run in Business Manager. In Business Manager go to Administration —> Operations —>Job Schedules. Select the job which you want to run:

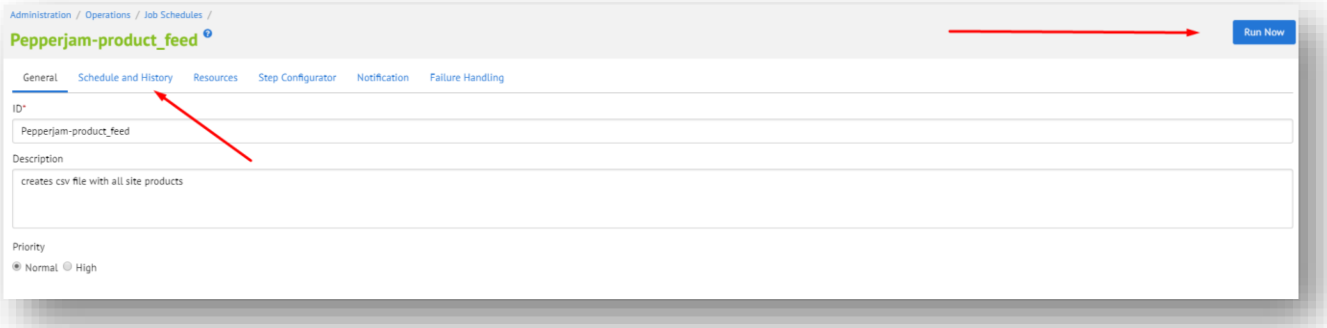


Administration / Operations /

Job Schedules

| □ | ID ▴ | Status | Last Run | Execution Scope | Resources | Priority | Enabled | Delete |
|---|---|---|---|---|---|---|---|---|
| □ | | - | | | - | | ⊘ | 🗑 |
| □ | Pepperjam-product_feed | OK | 5/30/2018 10:26 am | | - | | ⊘ | 🗑 |
| □ | Pepperjam_correction_order | OK | 6/4/2018 7:05 am | | - | | ⊘ | 🗑 |
| □ | | - | | | - | | ✓ | 🗑 |

Then click run manually or schedule it as you need in the Schedule and History tab.



Pepperjam-product_feed

ID*: Pepperjam-product_feed

Description: creates csv file with all site products

Priority: ● Normal ○ High

## 5 Testing

Once you are done in your sandbox environment and have tested it, replicate the changes to your production environment following SFCC best practices on replication: https://documentation.demandware.com/DOC1/index.jsp?topic=%2Fcom.demandware.dochelp%2FDataReplication%2FReplicationBestPractices.html

Then, let Pepperjam know the steps above have been completed. They will begin the testing process and verification of the pixel.

## 6 Support

Along the way, if you need assistance with your implementation please contact your Pepperjam account strategist or launch team representative.