

# Subscribe Pro

---

*SFRA Cartridge*

---



## Table of Contents

### Table of Contents

<b>1.</b>	<b>Summary</b>	<b>4</b>
<b>2.</b>	<b>Component Overview</b>	<b>5</b>
2.1	Functional Overview .....	5
2.2	Use Cases.....	5
2.3	Limitations, Constraints.....	6
2.4	Compatibility .....	6
2.5	Privacy, Payment .....	6
<b>3.</b>	<b>Implementation Guide</b>	<b>7</b>
3.1	Setup .....	7
	Initial Install .....	7
3.2	Configuration .....	13
	Update Web Service Credentials.....	13
	Setup Schedule Job to process orders with pending subscriptions.....	14
	Configure Subscription Products .....	16
	Configure OCAPI client .....	27
	Configure Payments .....	29
3.3	Custom Code.....	31
	Custom Code .....	33
3.4	External Interfaces .....	37
	Service Profile .....	37
	Service Credentials .....	38
	Service Endpoints .....	39
3.5	Testing.....	40
	Sandbox(es) .....	40
	Integration Testing .....	40
<b>4.</b>	<b>Operations, Maintenance</b>	<b>42</b>
4.1	Data Storage .....	42
4.2	Availability.....	42
4.3	Support.....	42
<b>5.</b>	<b>User Guide</b>	<b>43</b>
5.1	Roles, Responsibilities .....	43

<b>5.2</b>	<b>Business Manager .....</b>	<b>43</b>
<b>5.3</b>	<b>Storefront Functionality .....</b>	<b>44</b>
	Product Detail Page .....	44
	Cart Page .....	45
	Login Page .....	46
	Order Summary Right Rail and Minicart .....	47
	Order Confirmation Page .....	48
<b>6.</b>	<b><i>Known Issues</i></b>	<b>49</b>
<b>7.</b>	<b><i>Release History</i></b>	<b>50</b>

## 1. Summary

This document provides technical instructions for installing the Commerce Cloud Integration LINK integration that lets you integrate Subscribe Pro product subscriptions with a Commerce Cloud storefront.

This cartridge integrates Salesforce Commerce Cloud with the Subscribe Pro Subscription Commerce Platform.

This integration requires deployment of a new cartridge and modification of Storefront code.

A subscription to the Subscribe Pro solution is required to use this cartridge.  
Sandbox (testing) environments are available.

The provided “int\_subscribe\_pro\_sfra” cartridge is implemented using StoreFront Reference Architecture.

This integration is based on the SFRA Demo Store provided by Commerce Cloud.

## 2. Component Overview

### 2.1 Functional Overview

---

Subscribe Pro enables brands and retailers to add subscription commerce, auto-ship, continuity, clubs, boxes and recurring billing features to their existing eCommerce site, fully integrated with their existing eCommerce technology. Subscribe Pro supports recurring payments via credit card, ApplePay and eCheck, with pre-built integrations for all common payment gateways. Contact us today to find out how our solution can help you implement a product subscription program.

### 2.2 Use Cases

---

**Scenario:** A user visits a site that has subscription products available to purchase and would like to purchase those Products one time in order to try the product out before signing up for a recurring subscription.

**Result:** User will be presented with various options to subscribe to the subscription product with one of the options being a one time purchase. The user should be able to add this product to cart and checkout as if the product is a normal product in the catalog. Messaging should be present throughout checkout to inform the user that this is a one time purchase.

**Scenario:** A user visits a site that has subscription products available to purchase and would like to purchase a subscription product on a recurring interval automatically.

**Result:** User will be presented with various options to subscribe to the subscription product with one of the options being a subscription. The user should be able to add this product to cart and checkout as if the product is a normal product in the catalog. Messaging should be present throughout checkout to inform the user that this is a subscription purchase and display the interval they have chosen for repeat orders.

**Scenario:** A user has added some subscription products to their Basket and would like to adjust some of the values.

**Result:** User will be presented with options to change their subscription settings on the Cart Page. The user will also be able to click the “edit” link, on the subscription product, and change the subscription options in the quick view dialog.

**Scenario:** User has purchased a subscription product. The next time the ProcessSubproSubscriptions is run the necessary details about the customer and the products purchased will be transmitted to Subscribe Pro and the order will be marked as having been processed.

## 2.3 Limitations, Constraints

---

- Merchants must be enrolled at Subscribe Pro
- Subscription products must be configured at Subscribe Pro and Sales Force Commerce Cloud
- Apply Pay or a Credit Card will be required for subscription products
- Merchant must use Subscribe Pro as their Apple Pay Provider, if they want to use Apple Pay for Subscription Products
- Customers must be logged into the storefront in order to purchase subscription products
- Customer must save their Credit Card to their account in order to purchase subscription products

## 2.4 Compatibility

---

Compatible Demandware API version 16.2

Compatible SFRA Version: 3.3.0

## 2.5 Privacy, Payment

---

Customer profile data and payment information will be transferred and stored at Subscribe Pro.

This information includes:

- Name
- Email
- Address Details
- Saved Payment Instrument ID

Subscribe Pro will be handling Apple Pay Payment transactions. Subscribe Pro will store the Customers saved Payment Instrument ID, in order to process recurring orders.

## 3. Implementation Guide

### 3.1 Setup

---

Subscribe Pro LINK SFRA integration uses the following cartridges:

- *int\_subscribe\_pro\_sfra* – This cartridge contains the scripts, controllers, templates, metadata, and all other code necessary for implementing the Subscribe Pro integration with Salesforce Commerce Cloud SFRA.

Complete the following steps to install the cartridge.

#### *Initial Install*

---

##### ***Compilation***

Prior to uploading the Subscribe Pro SFRA cartridge you will need to compile the Javascript and CSS.

1. First create a directory called LINK\_subscribepro.
2. Next, inside this directory, clone the salesforceCommerceCloud/storefront-reference-architecture salesforceCommerceCloud/link\_subscribepro repositories
3. Make sure you have installed the sgmf-scripts module.
4. Run sgmf-scripts --compile js && sgmf-scripts --compile css

Next, you can upload your cartridge to your SFCC site.

5. Ensure a dw.json file exists with the credentials to upload to the correct store and deployment version.
6. Run sgmf-scripts --uploadCartridge int\_subscribe\_pro\_sfra

##### ***Update Site's cartridge path***

1. Navigate to *Administration > Sites > Manage Sites > [SiteName] > Settings tab*.
2. Add "int\_subscribe\_pro\_sfra:" in the cartridge path to the left of "app\_storefront\_base" in the effective cartridge path.
3. Click "Apply".

##### ***Install Services, Meta Data, and ProcessSubproSubscriptions Job***

1. First, package the contents of the Metadata directory (including the site\_import directory) into a Zip archive called site\_import.zip.
2. Navigate to *Administration > Site Development > Site Import & Export*
3. Upload the site\_import.zip archive and import it.

After the import is complete, the following system object definitions should include these additional

attributes.

### ***System object definitions***

<b>CustomerAddress</b>			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproAddressID	Subscribe Pro Address ID	String	This attribute will contain the address id that is returned from Subscription Pro upon creating this address in the Subscribe Pro system.
<b>CustomerPaymentInstrument</b>			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproPaymentProfileID	Subscribe Pro Payment Profile ID	String	This attribute will contain the payment profile id that is returned from Subscription Pro upon creating payment profile record in the Subscribe Pro system.
subproCCPrefix	Credit Card Prefix	String	This attribute will contain the first 6 digits of the customers Credit Card, to be sent to Subscribe Pro.
<b>Order / Basket</b>			
<i>Basket duplicates the order attributes so that they will automatically be passed from one to another.</i>			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproContainsSubscriptions	Contains Subscriptions	Boolean	This attribute will be used to quickly assess if an order has subscriptions contained within it
subproSubscriptionsToBeProcessed	Subscriptions to be processed	Boolean	This attribute will note whether or not this order has subscriptions that need to be processed. This value will be set to true if any products in the basket have subscriptions and will be set to false after those subscriptions have been processed
<b>OrderAddress</b>			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproAddressID	Subscribe Pro Address ID	String	This attribute will contain the address id that is returned from Subscription Pro upon creating this address in the Subscribe Pro system.
customerAddressID	SFCC Customer Address ID	String	This attribute will contain the SFCC customer address id. This information may



			be useful when a customer is updating subscription options after an order has been placed.
<b>Product</b>			
ID	Name	Type	Description
subproSubscriptionEnabled	Subscription Enabled	Boolean	This will decide whether or not the PDP should attempt to show subscription information
<b>ProductLineItem</b>			
ID	Name	Type	Description
subproSubscriptionOptionMode	Subscription Option Mode	String	This attribute will contain a value for whether or not if this product is subscription or a one time purchase. Possible values are: onetime_purchase subscription
subproSubscriptionSelectedOptionMode	Subscription Selected Option Mode	String	This attribute will contain subscription mode which was chosen on this ProductLineItem
subproSubscriptionInterval	Subscription Refill Interval	String	This attribute will contain the user selected time interval to refresh a subscription product
subproSubscriptionAvailableIntervals	Subscription Available Intervals	String	This attribute will contain a string with names of available subscription refill intervals, separated by comma
subproSubscriptionCreated	Subscription Created	Boolean	
subproSubscriptionDateCreated	Subscription Date Created	Date	
subproSubscriptionID	Subscription ID	String	This will be populated when the order processing occurs and the actual subscription is created.
subproSubscriptionDiscount	Subscription Discount	Double	This attribute will contain subscription discount offered on this product, as either a percentage or a fixed price, depending on the value of the subproSubscriptionIsDiscountPercentage property
subproSubscriptionIsDiscountPercentage	Subscription Is Discount Percentage	Boolean	This attribute will contain true if the discount on the product specified as a percentage or false if it is a fixed price amount.

<b>Profile</b>			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproCustomerID	Subscribe Pro Customer ID	String	This attribute will contain the customer id that is returned from Subscription Pro upon creating this customers records in the Subscribe Pro system.
<b>Site preferences</b> Please configure the following site preferences:			
<b>ID</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
subproEnabled	Subpro Enabled	Boolean	On/off switch for Subscription features.
subproApplePayLoginMsg	Subpro Apple Pay Login Message	String	Message informing customer that he/she needs to be logged in as a registered customer to use Apple Pay with SubPro Subscription products
subproCheckoutLoginMsg	Subpro Checkout Login Message	String	This attribute will contain a message that informs customer that he/she needs to be logged in as a registered customer to proceed checkout with SubPro Subscription products
subproCreditCardRequirementMsg	Subpro Credit Card Requirement Message	String	This attribute will contain an error message noting that a credit card is required for subscriptions
subproOrderProcessingErrorMail	Subscribe Pro Order Processing Error Mail	String	An email address that order subscriptions processing failures will be sent to
subproWidgetScriptUrl	Subscribe Pro Widget Script Url	String	URL to Subscribe Pro Widget Script
subproAddressWidgetScriptUrl	Subscribe Pro Address Book Widget Script Url	String	URL to the Subscribe Pro Address Book Widget Script
subproWalletWidgetScriptUrl	Subscribe Pro Wallet Widget Script Url	String	URL to the Subscribe Pro Wallet Widget Script
subproApiBaseUrl	Subscribe Pro API Base Url	String	Subscribe Pro API Base URL
subproAPICredSuffix	Suffix to append to the Subscribe Pro Web Service Credentials	String	Suffix to append to the Subscribe Pro Web Service Credentials
subproSubscriptionsWidgetConfig	Subscribe Pro Subscriptions Widget Custom Config	Text Area	JSON object string containing configuration for widget theme and display customization.
<b>Payment Card</b>			

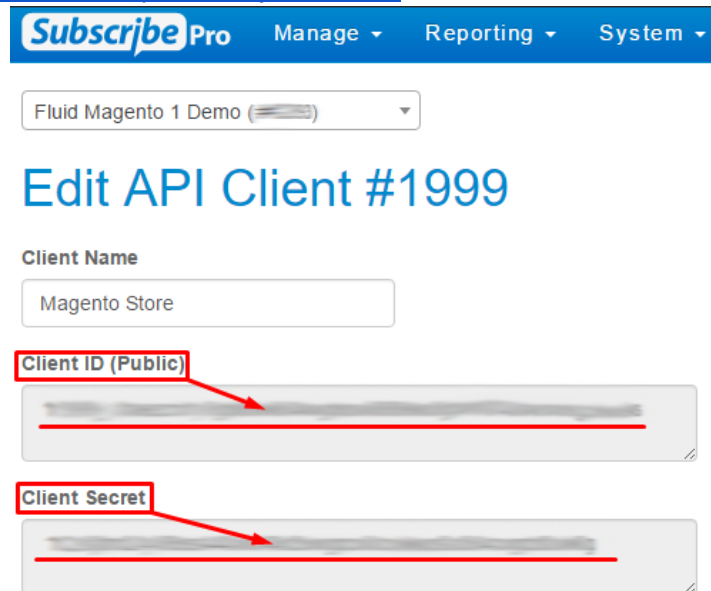
ID	Name	Type	Description
subproCardType	SubPro Card Type	String	Credit Card Type that matches the Credit Card Types at Subscribe Pro



## 3.2 Configuration

### Update Web Service Credentials

1. Get your Client ID and Client Secret in Subscribe Pro BM:  
<https://platform.subscribepro.com/system/client>



Subscribe Pro Manage Reporting System

Fluid Magento 1 Demo

## Edit API Client #1999

Client Name

Magento Store

Client ID (Public)

Client Secret

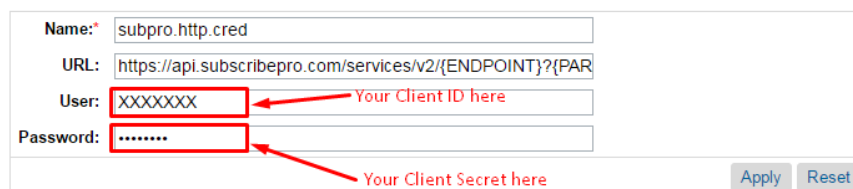
2. Navigate to *Administration > Operations > Services > Service Credentials > subpro.http.cred - Details*
3. Use Client ID as User and Client Secret as Password:

[Administration](#) > [Operations](#) > [Services](#) > [Service Credentials](#) > [subpro.http.cred - Details](#)

### subpro.http.cred

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 12 services.



Name: subpro.http.cred

URL: https://api.subscribepro.com/services/v2/{ENDPOINT}?{PAR

User: XXXXXXXX Your Client ID here

Password: \*\*\*\*\* Your Client Secret here

Apply Reset

4. Navigate to *Administration > Operations > Services > Service Credentials > subpro.http.cred.oauth - Details*
5. Use Client ID as User and Client Secret as Password (the same as above).

### *Turn on Subscribe Pro Cartridge Storefront Features*

---

Our cartridge adds an on/off switch in the configuration that allows you to enable or disable the frontend features of our integration.

Navigate to Merchant Tools > Site Preferences > Custom Preferences > Subscribe Pro, and find the Subscribe Pro Enabled setting. Set this to Yes to turn it on, or No to turn it off.

### *Setup Schedule Job to process orders with pending subscriptions*

---

The job will process all orders that have any subscriptions that need to be processed.

Configure job parameters:

1. Navigate to *Administration > Operations > Job Schedules > ProcessSubproSubscriptions* job.
2. On tab *Schedule and History* set run interval (default is every 15 minutes).
3. Open *Step Configurator* tab, choose *ProcessSubproSubscriptions-Process* workflow.
4. Set (**in hours**) custom parameter *ordersProcessInterval* to define what the start time for the search placed orders should be.
5. Click “Assign” button.
6. Additionally, check that the site scope is set properly.

## Configure Subscription Products

---

To configure products which will have a subscription you'll need to make changes to SFCC Business Manager as well as to Subscribe Pro Business Manager.

There are different type of products that can be configured:

- Standard Product
- Variation Product
- Variation Masters
- Product Bundles
- Product Sets
- Option Product

Below described action you need to perform to configure each type of product.

### Standard Product

Standard Product is a product that is sold and displayed alone and does not have variations, such as different sizes or colors. If you include a standard product in a product set or product bundle, it is considered a set product or bundled product, and not a standard product.

### Configuring Product on the Commerce Cloud Platform

The PDP page will take the Commerce Cloud Product ID and make an API request to the Subscribe Pro Products API Endpoint using the Commerce Cloud Product ID as the "sku" parameter.

The following image shows the Product ID that should be configured as a SKU in the Subscribe Pro Platform.

The screenshot shows the 'General' tab of a product configuration page in SFCC Business Manager. The breadcrumb trail at the top reads: Merchant Tools > Products and Catalogs > Products > 12345 - General. The product ID '12345' is highlighted with a red box. Below the ID, there is a warning message: 'You have not locked this product for editing. Click [Lock](#) if you need to edit the product.' Below this, there is a 'Select Language' dropdown menu set to 'Default'. Further down, the 'ID' field is set to '12345', the 'Catalog' is set to 'apparel-catalog', and the 'Tax Class' is set to 'Standard'. There are also fields for 'Searchable' (set to 'Default') and 'Searchable If Unavailable' (set to 'Default'), both with 'Yes' and '-None-' options respectively. Links for 'All Site Values' are provided for both searchability fields.



This API call will only be attempted if the Commerce Cloud Product has the: **Subscription Enabled** custom attribute set to: **true**.

The following image shows this setting in the following Business Manager Module:  
*Merchant Tools > Products and Catalogs > Products*

The screenshot displays the 'Products' configuration page in the Business Manager. The 'Subscribe Pro' section is highlighted with a red box, showing the 'Subscription Enabled' dropdown menu set to 'Yes'. Other visible sections include 'Refinement Color' (set to '-None-'), 'Skin Concern' (with options like 'dryTight', 'comprehensive', 'liftingLossFirm'), 'New Arrival?' (set to '-None-'), 'Sheets' (with options 'value1', 'value2', 'value3'), and 'On sale?' (set to '-None-'). Below these are sections for 'Clothing Attributes' (Style Number), 'Men's Accessories Attributes' (Size, Color), and a bottom navigation bar with 'Apply', 'Reset', 'Copy', 'Delete', and '<< Back to List' buttons.

### Configuring Product on the Subscribe Pro Platform

In addition to configuring the product on Commerce Cloud, the product will also need to be configured on the Subscribe Pro Platform.

<https://platform.subscribepro.com/login>

Subscription products can be managed on the platform at: *Manage > Subscription Products*

Within this module a new subscription product should be created that has a SKU that matches the Commerce Cloud Product ID.

The following is an example of a configured product that will match the product provided above for Commerce Cloud:

Fluid Magento 1 Demo (#2693)

## Edit Product #587218 - "Trial Test Product"

Product Settings Product Trial Settings

## Product Information

SKU Product Name  
12345 Trial Test Product

☒ Show Product On Subscribe Pro Management UI

## Subscription Widget - Subscription Option

## Subscription Option Mode

Subscription and One-Time Purchase

## Default Subscription Option

One-Time Purchase

## Product Options Handling

## Product Options Mode

Pass-through Options in Reorders

## Product Shipping Options

## Shipping Mode

Product Requires Shipping

## Product Pricing and Subscription Discount

## Full Price

\$ 100.00

☒ Discount By Percentage (Otherwise use Fixed Discount)

## Discount Percentage

20.00 %

## Fixed Discount Amount

\$ 0.00

## Subscription Quantities

## Min. Qty.

1

## Max. Qty.

9

## Product Intervals

Select which intervals shall be available for this product.

Drag the selected intervals into the order you wish them to appear in the subscription widget drop-down on your Magento site.

## Unselected

30 Days

45 Days

Daily

Every 2 Weeks

## Selected

Every 2 Months

Monthly

Weekly

## Default Interval

Select the interval which should be selected by default for this product.

[Back](#) [Save](#)[Documentation](#) | [Privacy](#) | [Terms](#)Copyright © 2012 - 2017 SUBSCRIBE PRO INC. All Rights Reserved.  
Subscribe Pro Platform Version 1.2.1.24

## ***Variation Product***

Variation Product is a product that is a specific variation. For example, if a Vegas brand t-shirt is represented by a variation master, then a product variation is a size 10, blue Vegas brand t-shirt. Variation products associated with the same variation master share most of the attributes defined for the product master, but have their own product IDs and images.

Variation Products should be configured the same way as a "Standard Product". See instructions above.

## ***Variation Masters***

Variation Masters is a product that represents all the variations for a particular product. For example, a specific style of t-shirt.

A variation master cannot be purchased directly, only product variations with specific variation information can be purchased. For example, if a master has color and size variation attributes, you must specify the color and size of a specific variant to purchase it.

If you have more than three variation attributes, you might want to consider using options.

To enable subscriptions for variation master products:

1. A Subscribe Pro Subscription Product will need to be created for each of the Variants contained within the Master Variant.
2. The master variant will need to have the "Subscriptions Enabled" custom attribute set to "yes".
  - a. Any attributes set on the Master Product will flow down to all of the variant products, unless the variant product has overridden that particular attribute.

**Example:**

### ***Master Product (in Commerce Cloud)***

ID	Name
25752235	Checked Silk Tie

### ***Variant Product(s) (in Commerce Cloud)***

Parent ID	ID	Options
25752235	682875090845	Color: Cobalt
25752235	682875719029	Color: Navy
25752235	682875540326	Color: Yellow

### ***Subscribe Pro - Subscription Products (in Subscribe Pro Platform)***

ID	SKU	Name
587218	682875090845	Checked Silk Tie - Cobalt
587219	682875719029	Checked Silk Tie - Navy

587220	682875540326	Checked Silk Tie - Yellow
--------	--------------	---------------------------

## ***Product Bundles***

Commerce Cloud Digital product bundles:

- Are a separate product with its own SKU and price
- Are unaffected by the price of the individual products
- Contain specified products in specified numbers

In respect to subscription products, product bundles can be treated the same as product variants. The end user will select subscription options for the entire bundle and the single product ID for the bundle is what will be configured and transmitted to Subscribe Pro when creating subscriptions. See the instructions how to configure a Standard Product above.

Sales Force Commerce Cloud will automatically add all of the bundled products (children) to the Basket when the product bundle is added, whether it be from the storefront or from OCAPI.

The following is an example of an OCAPI request and response using product bundle: sony-ps3-bundle:

Request
<pre>{   "customer_info": {     "customer_no": "00000001",     "email": "jkimmell@fluid.com"   },   "billing_address": {     "first_name": "James",     "last_name": "Kimmell",     "address1": "14137 Louise Drive",     "city": "Cumberland",     "state_code": "MD",     "country_code": "US"   },   "product_items": [{     "product_id": "sony-ps3-bundle",     "quantity": 1   }],   "shipments": [{     "gift": false,     "shipping_method": {       "id": "001"     },     "shipping_address": {       "first_name": "James",       "last_name": "Kimmell",       "address1": "14137 Louise Drive",       "city": "Cumberland",       "state_code": "MD",       "country_code": "US"     }   }] }</pre>

## Response

```
{
  "_v": "17.3",
  "_type": "basket",
  "_resource_state": "22e95720919f1eaa894bda606664b5a61a9575e0a942a1452bbfdb7062da0f69",
  "_flash": [
    {
      "_type": "flash",
      "type": "PaymentMethodRequired",
      "message": "No payment method ID was specified. Please provide a valid payment method ID.",
      "path": "$.payment_instruments[0].payment_method_id"
    }
  ],
  "adjusted_merchandize_total_tax": 22.45,
  "adjusted_shipping_total_tax": 0.5,
  "agent_basket": true,
  "basket_id": "16f3006604e77fff7105caf63a",
  "billing_address": {
    "_type": "order_address",
    "address1": "14137 Louise Drive",
    "city": "Cumberland",
    "country_code": "US",
    "first_name": "James",
    "full_name": "James Kimmell",
    "id": "b600c40aec47165cacc48c7ad2",
    "last_name": "Kimmell",
    "state_code": "MD"
  },
  "channel_type": "callcenter",
  "creation_date": "2017-04-19T15:26:23.314Z",
  "currency": "USD",
  "customer_info": {
    "_type": "customer_info",
    "customer_id": "abKT2hbWsyV8alhK6VJg90w0E",
    "customer_no": "00000001",
    "email": "jkimmell@fluid.com"
  },
  "last_modified": "2017-04-19T15:26:23.360Z",
  "merchandize_total_tax": 22.45,
  "notes": {
    "_type": "simple_link",
    "link": "https://subscribepro01-tech-prtnr-na02-dw.demandware.net/s/SiteGenesis/dw/shop/v17_3/baskets/16f3006604e77fff7105caf63a/notes"
  },
  "order_total": 481.94,
  "product_items": [
    {
      "_type": "product_item",
      "adjusted_tax": 22.45,
      "base_price": 449,
      "bonus_product_line_item": false,
      "bundled_product_items": [
        {
          "_type": "product_item",
          "adjusted_tax": null,
          "base_price": null,
          "bonus_product_line_item": false,
          "item_id": "94921b91f537474e71376abaf2",

```

```

        "item_text": "Sony Playstation 3 Game Console",
        "price": null,
        "price_after_item_discount": null,
        "price_after_order_discount": 0,
        "product_id": "sony-ps3-console",
        "product_name": "Sony Playstation 3 Game Console",
        "quantity": 1,
        "shipment_id": "me",
        "tax": null,
        "tax_basis": null,
        "tax_class_id": "standard",
        "tax_rate": 0.05
    },
    {
        "_type": "product_item",
        "adjusted_tax": null,
        "base_price": null,
        "bonus_product_line_item": false,
        "item_id": "6356afd70d4a120da36a9d3baf",
        "item_text": "Nascar 09 (for Sony PS3)",
        "price": null,
        "price_after_item_discount": null,
        "price_after_order_discount": 0,
        "product_id": "easports-nascar-09-ps3",
        "product_name": "Nascar 09 (for Sony PS3)",
        "quantity": 1,
        "shipment_id": "me",
        "tax": null,
        "tax_basis": null,
        "tax_class_id": "standard",
        "tax_rate": 0.05
    },
    {
        "_type": "product_item",
        "adjusted_tax": null,
        "base_price": null,
        "bonus_product_line_item": false,
        "item_id": "8b606524f0c7a7325cc4e4e23a",
        "item_text": "Monopoly Here and Now: The World Edition (for Sony PS3)",
        "price": null,
        "price_after_item_discount": null,
        "price_after_order_discount": 0,
        "product_id": "easports-monopoly-ps3",
        "product_name": "Monopoly Here and Now: The World Edition (for Sony PS3)",
        "quantity": 1,
        "shipment_id": "me",
        "tax": null,
        "tax_basis": null,
        "tax_class_id": "standard",
        "tax_rate": 0.05
    },
    {
        "_type": "product_item",
        "adjusted_tax": null,
        "base_price": null,
        "bonus_product_line_item": false,
        "item_id": "0798adadda95b0a5c4d282980f",
        "item_text": "Eternal Sonata (for Sony PS3)",
        "price": null,
        "price_after_item_discount": null,
        "price_after_order_discount": 0,
        "product_id": "namco-eternal-sonata-ps3",
    }

```

```

        "product_name":"Eternal Sonata (for Sony PS3)",
        "quantity":1,
        "shipment_id":"me",
        "tax":null,
        "tax_basis":null,
        "tax_class_id":"standard",
        "tax_rate":0.05
    },
    {
        "_type":"product_item",
        "adjusted_tax":null,
        "base_price":null,
        "bonus_product_line_item":false,
        "item_id":"b0bfeff3a8214ea603c570d73b",
        "item_text":"Warhawk (for Sony PS3)",
        "price":null,
        "price_after_item_discount":null,
        "price_after_order_discount":0,
        "product_id":"sony-warhawk-ps3",
        "product_name":"Warhawk (for Sony PS3)",
        "quantity":1,
        "shipment_id":"me",
        "tax":null,
        "tax_basis":null,
        "tax_class_id":"standard",
        "tax_rate":0.05
    }
],
"item_id":"8ddceb64e1d3856008a753122a",
"item_text":"Playstation 3 Bundle",
"option_items":[
    {
        "_type":"option_item",
        "adjusted_tax":0,
        "base_price":0,
        "bonus_product_line_item":false,
        "item_id":"392333b488e166df0b46927548",
        "item_text":"Extended Warranty: None",
        "option_id":"consoleWarranty",
        "option_value_id":"000",
        "price":0,
        "price_after_item_discount":0,
        "price_after_order_discount":0,
        "product_id":"000",
        "product_name":"Extended Warranty: None",
        "quantity":1,
        "shipment_id":"me",
        "tax":0,
        "tax_basis":0,
        "tax_class_id":"standard",
        "tax_rate":0.05
    }
],
"price":449,
"price_after_item_discount":449,
"price_after_order_discount":449,
"product_id":"sony-ps3-bundle",
"product_name":"Playstation 3 Bundle",
"quantity":1,
"shipment_id":"me",
"tax":22.45,
"tax_basis":449,

```

```

        "tax_class_id": "standard",
        "tax_rate": 0.05
    }
],
"product_sub_total": 449,
"product_total": 449,
"shipments": [
    {
        "_type": "shipment",
        "adjusted_merchandize_total_tax": 22.45,
        "adjusted_shipping_total_tax": 0.5,
        "gift": false,
        "merchandize_total_tax": 22.45,
        "product_sub_total": 449,
        "product_total": 449,
        "shipment_id": "me",
        "shipment_total": 481.94,
        "shipping_address": {
            "_type": "order_address",
            "address1": "14137 Louise Drive",
            "city": "Cumberland",
            "country_code": "US",
            "first_name": "James",
            "full_name": "James Kimmell",
            "id": "29410763e35c40915b515a0cdd",
            "last_name": "Kimmell",
            "state_code": "MD"
        },
        "shipping_method": {
            "_type": "shipping_method",
            "description": "Order received within 7-10 business days",
            "id": "001",
            "name": "Ground",
            "price": 9.99
        },
        "shipping_status": "not_shipped",
        "shipping_total": 9.99,
        "shipping_total_tax": 0.5,
        "tax_total": 22.95
    }
],
"shipping_items": [
    {
        "_type": "shipping_item",
        "adjusted_tax": 0.5,
        "base_price": 9.99,
        "item_id": "1ad26e6d3776f228e7d56228d7",
        "item_text": "Shipping",
        "price": 9.99,
        "price_after_item_discount": 9.99,
        "shipment_id": "me",
        "tax": 0.5,
        "tax_basis": 9.99,
        "tax_class_id": "standard",
        "tax_rate": 0.05
    }
],
"shipping_total": 9.99,
"shipping_total_tax": 0.5,
"taxation": "net",
"tax_total": 22.95
}

```



## Product Sets

Product Set is a product that includes several products that are displayed together and can be purchased together or separately. For example, an accessories kit that includes a hairbrush, comb, and mirror, each of which can also be bought separately. Another common example is an outfit, with a jacket, shirt, and pants, which can be purchased together or each item can be purchased separately.

Unlike Product Bundles, Products Set Products are added to the cart individually. The end consumer has the option to add each product individually or to add all of the products at once.

When the consumer adds all of the products at once, the form fields for each individual product are preserved, for example quantity and subscription options.

Products within a product set can be configured the same as a standard product and the user will have the ability to configure the subscription options and interval for each individual product. See the instructions for configuring a Standard Product above.

## Option Product

Commerce Cloud Digital, product options enable you to sell configurable products that have optional accessories, upgrades, or additional services. Options are always purchased with a product and cannot be purchased separately. They have their own price and display name, but do not have their own thumbnail images. They cannot be searched by the customer, but are usually visible on the product detail page.

Products that are configured with Product Options should be configured the same as a normal subscription product in Commerce Cloud and Subscribe Pro. The only difference with these products is that the Order Processing Script, that creates the subscription at Subscribe Pro, will also include any product options, selected for the product line item, in the `platform_specific_fields` attribute.

### Subscribe Pro Subscription Request

```
{
  "subscription":{
    "customer_id":"348328",
    "payment_profile_id":"410028",
    "requires_shipping":true,
    "shipping_address_id":455367,
    "product_sku":"canon-powershot-e1",
    "qty":1,
    "use_fixed_price":false,
    "interval":"Month",
    "next_order_date":"2017-04-19T21:33:56.954Z",
    "first_order_already_created":true,
    "send_customer_notification_email":true,
    "platform_specific_fields":{
      "sfcc":{
        "product_options":[
          {
            "id":"digitalCameraWarranty",
            "value":"002"
          }
        ]
      }
    }
  }
}
```

```

}
}
}
}
]
}

```

These product options will need to be added to the Basket when recurring orders are placed through OCAPI. The following is an example of a Basket object, with a product option, that could be sent back to Commerce Cloud via OCAPI:

## OCAPI Basket Object

```
{
  "customer_info": {
    "customer_no": "00000001",
    "email": "jkimmell@fluid.com"
  },
  "billing_address": {
    "first_name": "James",
    "last_name": "Kimmell",
    "address1": "14137 Louise Drive",
    "city": "Cumberland",
    "state_code": "MD",
    "country_code": "US"
  },
  "product_items": [{
    "product_id": "samsung-hl72a650",
    "quantity": 1,
    "option_items": [
      {
        "option_id": "tvWarranty",
        "option_value_id": "002"
      }
    ]
  }
],
  "shipments": [{
    "gift": false,
    "shipping_method": {
      "id": "001"
    },
    "shipping_address": {
      "first_name": "James",
      "last_name": "Kimmell",
      "address1": "14137 Louise Drive",
      "city": "Cumberland",
      "state_code": "MD",
      "country_code": "US"
    }
  }
]}
}
```

## Configure OCAPI client

---

Subscribe Pro will make use of the "Shop" Open Commerce API (OCAPI) in order to place recurring orders for subscription products. This will be accomplished server to server using a Business Manager login.

Configure client's resources as following:

```
{
  "_v": "17.3",
  "clients": [
    {
      "client_id": "aaaaaaaaaaaaaaaaaaaaaaaaaaaa",
      "allowed_origins": [],
      "resources": [
        {
          "resource_id": "/baskets",
          "methods": ["post"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/baskets/*",
          "methods": ["get", "patch", "delete"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/baskets/*/items",
          "methods": ["post"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/baskets/*/payment_instruments",
          "methods": ["post"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/baskets/*/price_adjustments",
          "methods": ["post"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/baskets/*/shipments/me",
          "methods": ["patch"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/customers/*/baskets",
          "methods": ["get"],
          "read_attributes": "(**)",
          "write_attributes": "(**)"
        },
        {
          "resource_id": "/customers/*",
          "methods": ["post"],
          "read_attributes": "(**)",

```

```

        "write_attributes": "(**)"
    },
    {
        "resource_id": "/products/*",
        "methods": ["get"],
        "read_attributes": "(**)",
        "write_attributes": "(**)"
    },
    {
        "resource_id": "/baskets/*",
        "methods": ["delete"],
        "read_attributes": "(**)",
        "write_attributes": "(**)"
    },
    {
        "resource_id": "/baskets/*/shipments/me/shipping_method",
        "methods": ["put"],
        "read_attributes": "(**)",
        "write_attributes": "(**)"
    },
    {
        "resource_id": "/orders",
        "methods": ["post", "get"],
        "read_attributes": "(**)",
        "write_attributes": "(**)"
    },
    {
        "resource_id": "/orders/*/payment_instruments/*",
        "methods": ["patch"],
        "read_attributes": "(**)",
        "write_attributes": "(**)"
    }
]
}

```

## Configure Payments

---

Subscribe Pro will make use of OCAPI to place orders on behalf of the customer. This is one of the reasons that a customer must be logged in and have a stored payment instrument in order to place an order with subscriptions.

### Credit Cards

A Payment token and the Payment Instrument ID will be transmitted to Subscribe Pro as part of the Subscription creation process.

Credit Card tokenization will need to be implemented by a tokenization provider when this cartridge is implemented, on an actual site. This will be accomplished by making use of the: `setCreditCardToken` method in the `PaymentInstrument` object.

See SFCC documentation:

[https://documentation.demandware.com/DOC1/topic/com.demandware.dochelp/DWAPI/scriptapi/html/api/class\\_dw\\_order\\_PaymentInstrument.html](https://documentation.demandware.com/DOC1/topic/com.demandware.dochelp/DWAPI/scriptapi/html/api/class_dw_order_PaymentInstrument.html)

### Subscribe Pro - Payment Profile

Payment profile should be used to pay for subscriptions.

The use of this profile could be divided into two steps:

- creating subscription
- recurring subscription charging

### Creating subscription

As part of the order processing script, we send the relevant Payment Data to the Subscribe Pro using the `/vault/paymentprofile/external-vault end-point`.

The following is the relevant snippet of code in the Order Processing Logic:

`/int_subscribe_pro_sfra/cartridge/scripts/subpro/jobs/ProcessOrderSubscriptions.js`

```
/**
 * Apple Pay otherwise assume Credit Card
 */
if (paymentInstrument.getPaymentMethod() === "DW_APPLE_PAY") {
    let transactionID = paymentInstrument.getPaymentTransaction().getTransactionID();
    let response = SubscribeProLib.getPaymentProfile(null, transactionID);

    /**
     * If there was a problem creating the payment profile, error out
     */
    if (response.error) {
        logError(response, 'getPaymentProfile');
        continue;
    } else {
        paymentProfileID = response.result.payment_profiles.pop().id;
    }
}
```

```

} else {
    paymentProfileID = ('subproPaymentProfileID' in customerPaymentInstrument.custom) ?
customerPaymentInstrument.custom.subproPaymentProfileID : false;

    /**
     * If Payment Profile already exists,
     * Call service to verify that it still exists at Subscribe Pro
     */
    if (paymentProfileID) {

        let response = SubscribeProLib.getPaymentProfile(paymentProfileID);

        /**
         * Payment Profile not found, create new Payment Profile record
         * Otherwise create the payment profile
         */
        if (response.error && response.result.code === 404) {
            paymentProfileID = createSubproPaymentProfile(customerProfile,
customerPaymentInstrument, order.billingAddress);
            /**
             * Some other error occurred, error out
             */
        } else if (response.error) {
            logError(response, 'getPaymentProfile');
            continue;
        }
    } else {
        paymentProfileID = createSubproPaymentProfile(customerProfile,
customerPaymentInstrument, order.billingAddress);
    }
}
}

```

### *Recurring subscription charging*

To charge a regular fee we should create OCAPI order using payment profile information saved earlier. Then using this data SubPro side will create order for paying for subscription via OCAPI.

1. dw/shop/{version}/baskets
  - a. This contains payment\_instruments, customer\_info, and billing\_address elements
2. dw/shop/{version}/baskets/{id}/shipments/me
  - a. To set the shipping method and address
3. dw/shop/{version}/baskets/{id}/items
  - a. Loops for each product added to the cart
4. dw/shop/{version}/baskets/{id}/price\_adjustments
  - a. Only called if a subscription discount is in effect, if so it is called along with each /items call
5. dw/shop/{version}/orders
  - a. Places the order
6. dw/shop/{version}/orders/{order\_id}/payment\_instruments/{order\_payment\_instrument\_id}
  - a. Triggers the payment authorization hooks

### 3.3 Custom Code

---

Subscribe Pro's SFRA cartridge does not directly modify code in the app\_storefront\_base cartridge. Instead, we follow best practices for extending and overriding the default code.

#### Changed Template Files – within int\_subscribe\_pro\_sfra/cartridge/templates/default

- account/
  - payment/
    - payment.isml – Adds Javascript code to configure and load Wallet widget
  - addressBook.isml – Adds Javascript code to configure and load Address Book widget
  - dashboardProfileCards.isml – Adds a profile card to the My Account section to allow the customer to access the My Subscriptions page/widget.
- cart/
  - productCard
    - cartBundleCard.isml – Adds cart subscription widget for bundle product line items
    - cartProductCard.isml – Adds cart subscription widget for product line items
  - cart.isml – Adds required javascript to the cart page
- checkout/
  - billing/
    - paymentOptions/
      - paymentOptionsContent.isml – Hide non credit card methods for subscription orders
      - paymentOptionsSummary.isml - Hide non credit card methods for subscription orders
      - paymentOptionsTabs.isml – Hide non credit card methods for subscription orders.
  - productCard/
    - productCard.isml – Show subscription details in order summary, mini cart, and order confirmation
  - checkout.isml – Include necessary javascript on checkout page
  - checkoutLogin.isml – Hide guest checkout option when subscription is in the cart
- product/
  - bundleDetails.isml – Show subscription widget on product bundle details page.
  - productDetails.isml – Show subscription widget on product details page.

#### Added Template Files

- subpro/
  - account/
    - mysubscriptions.isml – My Subscriptions page for customer subscription self-management
    - mysubscriptionsCard.isml – Card for linking the customer to My Subscriptions page from My Account page
  - cart/
    - subprooptions.isml - Subscription widget for cart page
  - mail/
    - orderprocessingerror.isml – Email template for order processing error
  - product/
    - subprooptions.isml – Subscription widget for product page

## Controllers

- Address.js – Added code to handle syncing address creation/modification/deletion through Address Book Widget
- Checkout.js – Return details about whether a subscription is in the cart to force login for subscription orders
- PaymentInstruments.js – Added code to handle syncing payment instrument creation/modification/deletion through Wallet Widget
- SubPro.js – Returns view data for and handles user interaction with subscription widget in PDP and cart
- Subscriptions.js – My Subscriptions Widget page.

## Client Scripts

- client/default/js/cart.js – Include subscriptionOptions.js for the cart
- client/default/js/productDetail.js – Include subscriptionOptions.js for the product page
- client/default/js/subscriptionOptions.js – Clientside JS for subscription widget and checkout process
- client/default/js/addressBook.js – Include custom client-side javascript on the address book page
- client/default/js/addressBook/addressBook.js – Custom client-side javascript for the address book page
- client/default/js/spPaymentInstruments.js – Include spPaymentInstruments/spPaymentInstruments.js on payment instruments page.
- client/default/js/spPaymentInstruments/spPaymentInstruments.js – Client-side javascript to reload payment instruments page after deleting a payment instrument, to trigger sync to Subscribe Pro through Wallet Widget.

## Server Scripts

- scripts/subpro/helpers/\* - Helpers for address, customer, payment, and widget entities
- scripts/subpro/hooks/applePay.js – ApplePay integration hook
- scripts/subpro/init/httpServiceInit.js – Init script for API services
- scripts/job/ProcessOrderSubscriptions.js – Job to sync subscriptions into Subscribe Pro after initial subscription order is placed
- scripts/subpro/mock\_data/\* - Mock data to be used in tests or as examples

## Hooks

- hooks.json

## Resource

- account.properties
- order.properties
- product.properties

Below are the changes that were made to several of the app\_storefront\_base files that are included in our cartridge to override the original files and include Subscribe Pro functionality. These changes are included with cartridge and are displayed here only for informational purposes to aid with further customization, or to applying the changes to a custom theme.



**Modify account/payment/payment.isml**

First, in the isscript tag at the top, add the following:

```
assets.addJs('/js/spPaymentInstruments.js');
```

Next, within the div.container tag, after the isloop tag, add the following:

```
<!-- Load the Subscribe Pro widget script -->
<script
src="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('subproWalletWidgets
criptUrl')}"></script>

<div id="sp-wallet-assist-widget"></div>

<isif condition="${pdict.newCard != 'null' || pdict.deletedCard != 'null' }">
  <isset name="widgetConfig" scope="page"

value="${require('~cartridge/scripts/subpro/helpers/WidgetsHelper').getWidgetConfig(custom
er.profile.custom.subproCustomerID, 'client_credentials', 'widget', 'sp-wallet-assist-
widget')}" />
    <!-- Pass configuration and init the Subscribe Pro widget -->
    <script>
      // Setup config for Subscribe Pro
      var widgetConfig = <isprint value="${JSON.stringify(widgetConfig)}"
encoding="off" />;
      WalletAssist.init(widgetConfig);
      <isif condition="${pdict.newCard != 'null'}">
        var newCard = <isprint value="${pdict.newCard}" encoding="off" />;
        WalletAssist.onPaymentProfileCreated(newCard, function (result) {
          var newCardSfccId = <isprint value="${pdict.newCardSfccId}"
encoding="off" />;
          jQuery.get("${URLUtils.url('PaymentInstruments-
SetSPPaymentProfileID')}", {"paymentInstrumentId": newCardSfccId, "spPaymentProfileId":
result.id});
        });
      </isif>
      <isif condition="${pdict.deletedCard != 'null'}">
        var deletedCard = <isprint value="${pdict.deletedCard}" encoding="off"
/>;
        WalletAssist.onPaymentProfileDeleted(deletedCard);
      </isif>
    </script>
  </isif>
  <div class="row justify-content-center">
    <div class="col-6">
      <div class="row">
        <div class="col">
          <a href="${URLUtils.url('Account-Show')}">
            <p class="text-center back-to-account-link" aria-
label="${Resource.msg('label.addressbook.myaccount', 'account', null)}">${Resource.msg('link.
profile.backtomyaccount', 'account', null)}</p>
          </a>
        </div>
        <div class="col">
          <a href="${URLUtils.url('PaymentInstruments-AddPayment')}"
class="btn btn-save btn-block btn-primary" aria-
label="${Resource.msg('label.payment.addnewpayment', 'payment', null)}">${Resource.msg('butto
n.addnew', 'payment', null)}</a>
        </div>
```

```

        </div>
    </div>
</div>

```

### Modify account/addressBook.isml

After the isloop tag, add the following:

```

        <isif condition="${pdict.newAddress != 'null' || pdict.updatedAddress != 'null' ||
pdict.deletedAddress != 'null'}">
            <!-- Subscribe Pro Address Book Widget -->
            <script
src="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('subproAddressWidget
ScriptUrl')}"></script>
            <div id="sp-address-book-widget"></div>
            <isset name="widgetConfig" scope="page"

value="${require('~cartridge/scripts/subpro/helpers/WidgetsHelper').getWidgetConfig(custom
er.profile.custom.subproCustomerID, 'client_credentials', 'widget', 'sp-address-book-
widget')}"/>
            <!-- Pass configuration and init the Subscribe Pro widget -->
            <script>
                // Setup config for Subscribe Pro
                var widgetConfig = <isprint value="${JSON.stringify(widgetConfig)}"
encoding="off" />;
                AddressBookAssist.init(widgetConfig);
                <isif condition="${pdict.newAddress != 'null'}">
                    var newAddress = <isprint value="${pdict.newAddress}" encoding="off"
/>;
                    var newAddressSfccId = '<isprint value="${pdict.newAddressSfccId}"
encoding="off" />';
                    AddressBookAssist.onAddressCreated(newAddress).then(function (result) {
                        jQuery.get("${URLUtils.url('Address-SetSPAddressID')}"),
                        {"addressId": newAddressSfccId, "spAddressId": result.id});
                    });
                </isif>
                <isif condition="${pdict.updatedAddress != 'null'}">
                    var updatedAddress = <isprint value="${pdict.updatedAddress}"
encoding="off" />;
                    AddressBookAssist.onAddressUpdated(updatedAddress);
                </isif>
                <isif condition="${pdict.deletedAddress != 'null'}">
                    var deletedAddress = <isprint value="${pdict.deletedAddress}"
encoding="off" />;
                    AddressBookAssist.onAddressDeleted(deletedAddress);
                </isif>
            </script>
        </isif>

        <div class="row justify-content-center">
            <div class="col-6">
                <div class="row">
                    <div class="col text-center">
                        <a href="${URLUtils.url('Account-Show')}" class="normal" aria-
label="${Resource.msg('label.addressbook.myaccount', 'account', null)}">
                            ${Resource.msg('link.profile.backtomyaccount', 'account', null)}
                        </a>
                    </div>
                    <div class="col">
                        <a href="${URLUtils.url('Address-AddAddress')}" class="btn btn-save
btn-block btn-primary" aria-
label="${Resource.msg('label.addressbook.addnewaddress', 'account', null)}">${Resource.msg('b
utton.addnew', 'account', null)}</a>
                    </div>
                </div>
            </div>
        </div>

```

```
        </div>
      </div>
    </div>
  </div>
```

### ***Modify account/dashboardProfileCards.isml***

After the `isinclude` for `paymentCard`, add the following:

```
<!--Subscriptions-->
<isinclude template="subpro/account/mysubscriptionsCard"/>
```

### ***Modify product/bundleDetails.isml***

After the `isinclude` for `Availability`, add the following:

```
<!-- Subscription Widget -->
<isinclude url="{URLUtils.url('SubPro-PDP', 'sku', product.id)}" />
```

### ***Modify product/productDetails.isml***

Add the following below the `Applicable Promotions` include.

```
<!-- Subscription Widget -->
<div class="col-md-10 col-12">
  <isinclude url="{URLUtils.url('SubPro-PDP', 'sku', product.id)}" />
</div>
```

### ***Modify cart/productCard/cartBundleCard.isml***

Before the `isinclude` for `cartProductCardProductPrice`, add this:

```
<!-- Subscribe Pro - Subscription Details for Mini Cart, Order Summary, and Order Confirmation -->
<isinclude url="{URLUtils.url('SubPro-Cart', 'sku', lineItem.id)}" />
```

### ***Modify cart/productCard/cartProductCard.isml***

After the `isinclude` for `cartProductCardEdit`, after the next two closing `div` tags, add the following:

```
<!-- Subscribe Pro - Subscription Details for Mini Cart, Order Summary, and Order Confirmation -->
<div class="col-md-12">
  <isinclude url="{URLUtils.url('SubPro-Cart', 'sku', lineItem.id)}" />
</div>
```

### ***Modify cart/cart.isml***

Add the following inside the `isscript` tag at the top:

```
assets.addJs('/js/cart.js');
```

### ***Modify checkout/productCard/productCard.isml***

Add the following after the include of cartProductCardAvailability.

```
<!-- Subscribe Pro - Subscription Details for Mini Cart, Order Summary, and Order Confirmation ->
<isif condition="${miniCart || pdict.order.orderNumber == null}">
    <isinclude url="${URLUtils.url('SubPro-OrderSummary', 'sku', lineItem.id)}" />
</isif>
<isif condition="${pdict.order && pdict.order.orderNumber != null}">
    <isinclude url="${URLUtils.url('SubPro-OrderConfirmation', 'sku', lineItem.id,
'orderId', pdict.order.orderId)}" />
</isif>
```

### ***Modify checkout/checkout.isml***

Add the var SubscribeProLib = ... line, and the <isset name="isSubPro" ...> lines as shown below.

```
<isscript>
    var assets = require('*/cartridge/scripts/assets.js');
    var SubscribeProLib = require('*/cartridge/scripts/subpro/lib/SubscribeProLib.js');
    assets.addJs('/js/checkout.js');
    assets.addCss('/css/checkout/checkout.css');
</isscript>

<isset name="isSubPro" value="${SubscribeProLib.isSubPro()}" scope="page" />
```

### ***Modify checkout/checkoutLogin.isml***

The following change removes the Guest Checkout option when a subscription is in the cart.

Locate template `/int_subscribe_pro_sfra/cartridge/templates/default/checkout/checkoutLogin.isml`. Find the `div.card` that contains the link to begin guest checkout, and put the following `isif/iselse` statement around it.

```
<!-- Hide guest checkout when a subscription item is in the cart -->
<isif condition="${!pdict.subscriptionInCart}">
    <div class="card">
        <div class="card-header">
            <h4>${Resource.msg('heading.guest.checkout', 'checkout', null)}</h4>
        </div>
        <div class="card-body">
            <p>
                ${Resource.msg('msg.guest.checkout1', 'checkout', null)}
                ${Resource.msg('msg.guest.checkout2', 'checkout', null)}
            </p>
            <a
                href="${URLUtils.https('Checkout-Begin')}"
                class="btn btn-block btn-primary checkout-as-guest">
                ${Resource.msg('button.checkout.as.guest', 'checkout', null)}
            </a>
        </div>
    </div>
```

```

    </div>
    <iselse>

    ${require('dw/system/Site').getCurrent().getCustomPreferenceValue('subproCheckoutLoginMsg')}
    }
  </isif>

```

### ***Modify checkout/billing/paymentOptions/paymentOptionsTabs.isml, paymentOptionsSummary.isml, paymentOptionsContent.isml***

The following change to each of these three files removes the non-credit card payment methods during checkout when a subscription item is in the cart.

Locate these templates:

- */int\_subscribe\_pro\_sfra/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsTabs.isml*
- */int\_subscribe\_pro\_sfra/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsSummary.isml*
- */int\_subscribe\_pro\_sfra/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsContent.isml*

Add the following iselseif tag within the condition after the isinclude.

```

<iselseif condition="${isSubPro}">
  <iscontinue/>
</iselseif>

```

## **3.4 External Interfaces**

Web service calls made to the Subscribe Pro RESTful API will make use of Sales Force Commerce Cloud Services:

<https://documentation.demandware.com/DOC1/topic/com.demandware.dochelp/WebServices/Webservices.html>

Application code should not call the web services directly but should instead call methods within the Subscribe Pro Javascript module:

*/int\_subscribe\_pro\_sfra /cartridge/scripts/subpro/lib/SubscribeProLib.js*

### ***Service Profile***

There are separate profiles for GET vs POST request to enable / disable the circuit breaker.

<b>Name</b>	subpro.http.prof.get	subpro.http.prof.post
<b>Description</b>	Profile for GET requests	Profile for POST requests

<b>Connection Timeout (ms)</b>	5000	5000
<b>Enable Circuit Breaker</b>	yes	no
<b>Max Circuit Breaker Calls</b>	3	0
<b>Enable Rate Limit</b>	1000	1000
<b>Max Rate Limit Calls</b>		
<b>Rate Limit Interval (ms)</b>		

### *Service Credentials*

---

The *"siteName"* portion of the Service Credential comes from the Site Preference: *subproAPICredSuffix*. This allows merchants to have different credentials for each site.

The default value of the Site Preference is *siteName*, so that it will automatically match the default credentials.

Please note that the user password will need to be updated after import.

<b>Name</b>	subpro.http.cred.siteName
<b>URL</b>	https://api.subscribepro.com/services/v2/{ENDPOINT}?{PARAMS}
<b>User</b>	Subscribe Pro Client ID
<b>Password</b>	Subscribe Pro Client Secret

<b>Name</b>	subpro.http.cred.oauth.siteName
<b>URL</b>	https://api.subscribepro.com/oauth/v2/{ENDPOINT}?{PARAMS}
<b>User</b>	Subscribe Pro Client ID
<b>Password</b>	Subscribe Pro Client Secret

## Service Endpoints

---

ID	API endpoint	Description
subpro.http.get.config	GET /services/v2/config	Retrieve configuration for the authenticated account
subpro.http.get.subscriptions	GET /services/v2/subscriptions	Retrieve a collection of subscriptions.
subpro.http.post.subscription	POST /services/v2/subscription	Create a new subscription
subpro.http.post.addresses	POST /services/v2/address	Create / Update a Subscribe Pro addresses
subpro.http.post.addressfindcreate	POST /services/v2/address/find-or-create	Find a matching address or create a new one
subpro.http.get.addresses	GET /services/v2/addresses	Retrieve a collection of addresses
subpro.http.get.products	GET /services/v2/products/{id}	Retrieve a single product by id
subpro.http.get.customers	GET /services/v2/customers/{id}	Retrieve a single customer by id
subpro.http.post.customer	POST /services/v2/customer	Create a new customer
subpro.http.post.customers	POST /services/v2/customers/{id}	Update a single customer, identified by customer id
subpro.http.get.token	GET /oauth/v2/token	Get an OAuth2 access token
subpro.http.get.paymentprofile	GET /services/v2/vault/paymentprofiles/{id}	Retrieve a single payment profile by id
subpro.http.post.paymentprofile.vault	POST /services/v2/vault/paymentprofile/external-vault	Create a new payment profile for an external vault

For more details visit Subscribe Pro REST API Documentation:

<https://platform.subscribepro.com/docs/rest>

## 3.5 Testing

---

### *Sandbox(es)*

---

In order to test the application in a non-production environment, you will need to have a Subscribe Pro Sandbox.

Subscribe Pro provides "sandbox" accounts to all merchants.

- One Sandbox account is included with standard plans
- Additional sandbox accounts (or those which aren't associated with a paid plan) can be added but will incur a fee.

### *Integration Testing*

---

#### **Subscription Products**

The following are some test cases that can be completed to ensure subscription products are configured correctly.

1. User's should see subscription options for products, on the product detail page, that have subscription's enabled and a corresponding product has been configured on the Subscribe Pro Platform.
2. User's should see an option for a one time purchase, if this option has been configured on Subscribe Pro.
3. User's should see an option to purchase a product on a recurring basis, if this option has been configured on Subscribe Pro.
4. User's should see the configured repeat Order options and be able to select the desired frequency of recurring orders.
5. User's should be able to add a product, with subscription options, to the Basket and see the selected options present in the: Mini-Cart, Basket, Order Details (in the right rail of Checkout), the Order Confirmation Screen and in the Order Confirmation Email.
6. User's should be able to change their subscription options while viewing the product in the Basket.
7. Subscription details should be transmitted to Subscribe Pro through a configured Job / Job Schedule.
8. Failed requests to create a subscription should be emailed to the configured user for further investigation.



## **My Account**

The following are some test cases that can be completed to ensure that the My Account has been configured correctly.

1. User's should be able to see a list of their Subscription orders in the "My Subscriptions" section of "My Account".
2. User's should see an option to update their Subscription Payment option upon changing their saved payment instrument in "My Account".
3. User's should see an option to update their Subscription Address option upon changing their saved address in "My Account".
4. User's should be able to update their profile information in "My Account" and those details are automatically transmitted to Subscribe Pro.

## **Pixel / Widget Tracking**

The following are some test cases that can be completed to ensure that the Subscribe Pro Widget is included in all of the necessary locations.

1. The Javascript Widget should be executed and transmitting details to Subscribe Pro via Ajax on the following pages:
  - a. My Account – Addresses
  - b. My Account – Payment Instruments
  - c. My Account – My Subscriptions

## 4. Operations, Maintenance

### 4.1 Data Storage

---

Subscription selections and Subscribe Pro identifiers (customer, address, subscription) will be stored to the custom attributes of the Sales Force Commerce Cloud System Objects (Product, Product Line item, Order, Basket).

Customer information such as name, email, address and some payment information will be stored at Subscribe Pro for any customers that have purchased subscription products. This information will be used to process future recurring orders for the customer.

### 4.2 Availability

---

Subscribe Pro hosted services and API's perform with an uptime of 99.9% reliability.

If external services are not available for products, subscription options will not be available for subscription products. The end user will still be able to purchase products as normal on the platform but subscription options will not be available.

If external services are not available for My Account, the end user will see a My Subscription page without any content.

If external services are not available for transmitting subscription order details, via a scheduled job, an email should be sent to the configured contact and they will need to manually process the subscription.

### 4.3 Support

---

Support will be provided through the Subscribe Pro support system.

All requests should include: details of issue, email, phone, and URL.

Requests can be made at the following location:

<https://www.subscribepro.com/support/> or by emailing [support@subscribepro.com](mailto:support@subscribepro.com)

**Technical Support Contact:**

David King

[david.king@subscribepro.com](mailto:david.king@subscribepro.com)

## 5. User Guide


### 5.1 Roles, Responsibilities

<LIST RECURRING TASKS THAT NEED TO BE FULFILLED BY CUSTOMER, MERCHANT TO RUN THE INTEGRATION, E.G. MANUAL FEED OF CATALOG DATA INTO 3RD PARTY SERVICE, IF APPLICABLE>

### 5.2 Business Manager

Subscribe Pro group is added to Custom Site Preference Groups (*Site > Site Preferences > Custom Preferences*). It contains following attributes:

[Merchant Tools](#) / [Site Preferences](#) / [Custom Site Preference Groups](#) /

**Subscribe Pro** 

Cancel


Apply to Other Sites


Save

Instance Type

Sandbox

Search by IDs...





1-10

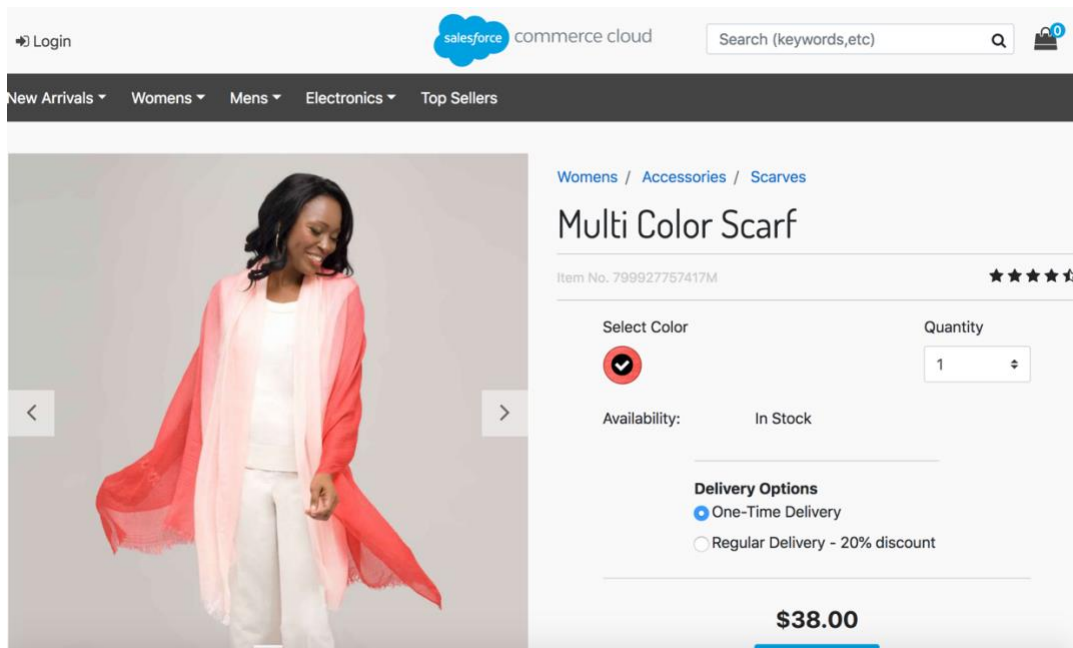
of 10

Name	Value	Default Value	
Subscribe Pro Enabled	<div>Yes</div>	Yes	<a href="#">Edit Across Sites</a>
Subpro Credit Card Requirement Message	<div><div></div><div>Error message noting that a credit card is required for subscr...</div></div>	You need a credit card to pay fo...	<a href="#">Edit Across Sites</a>
Subpro Checkout Login Message	<div><div></div><div>Message informing customer that he/she needs to be logge...</div></div>	You need to be logged in as a r...	<a href="#">Edit Across Sites</a>
Subpro Apple Pay Login Message	<div><div></div><div>Message informing customer that he/she needs to be logge...</div></div>	You need to be logged in as a r...	<a href="#">Edit Across Sites</a>
Subscribe Pro Order Processing Error Mail	<div><div></div><div>An email address that order subscriptions processing failure...</div></div>		<a href="#">Edit Across Sites</a>
Suffix to append to the Subscribe Pro Web Service Credentials	<div>RefArch</div>	siteName	<a href="#">Edit Across Sites</a>
Subscribe Pro Widget Script Url	<div>https://static.subscribepro.com/my-subscriptions/widget-1</div>	https://widgets.subscribepro.co...	<a href="#">Edit Across Sites</a>
Subscribe Pro API Base Url	<div>https://api.subscribepro.com</div>	https://api.subscribepro.com	<a href="#">Edit Across Sites</a>
Subscribe Pro Wallet Widget Script URL	<div>https://static.subscribepro.com/wallet/widget-wallet-1.0.1</div>		<a href="#">Edit Across Sites</a>
Subscribe Pro Address Book Widget Script URL	<div>https://static.subscribepro.com/address-book/widget-addi</div>		<a href="#">Edit Across Sites</a>

## 5.3 Storefront Functionality

### Product Detail Page

When the product page template has been updated, you should be able to see subscription options for product: one time delivery or regular delivery and drop down with delivery intervals if regular delivery is available. This information will be shown only for products for which subscription is configured in SFCC Business Manager and in Subscribe Pro Business Manager.




## Cart Page

If any product with subscription was added to basket, on cart page you will see the same subscription options as on PDP for such product line item. At this step subscription options can be changed. To apply changes you'll need to click on 'Update Cart' button.

# Your Cart

[Continue Shopping](#)1 Items[Need Help? Call 1-800-555-0199](#)

### Multi Color Scarf



Color: Green  
In Stock

**Each**  
**\$38.00**

**Quantity**  
1

**Total**  
~~\$38.00~~  
**\$30.40**

**Delivery Options**

☐ One-Time Delivery

☒ Regular Delivery -  
20% discount

Delivery every

Monthly

[Edit](#)

Enter Promo Code

Promo Code [Submit](#)

Shipping

Ground ( 7-10 Business Days )

Shipping cost \$5.99

Sales Tax \$1.82

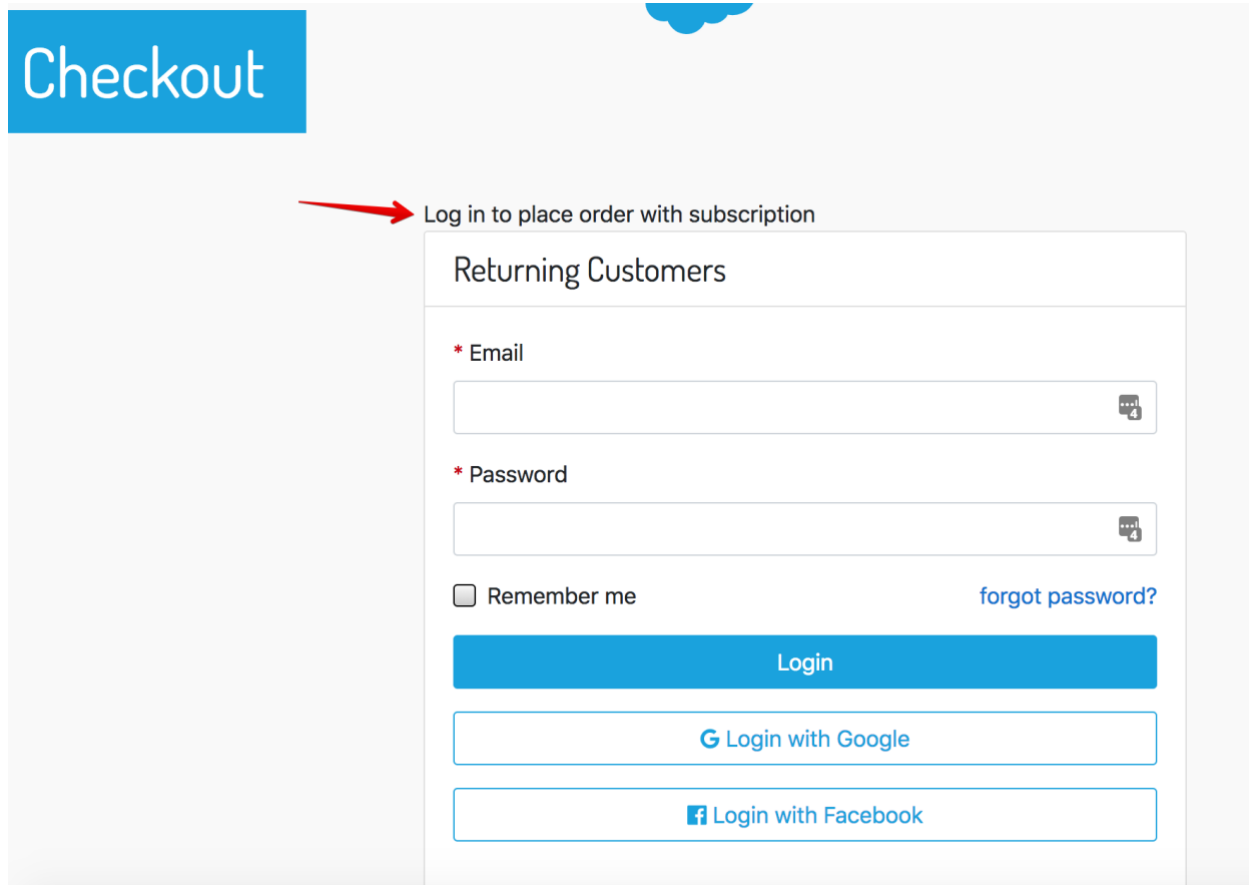
**Estimated Total** **\$38.21**

[Checkout](#)

## Login Page

---

If customer has any product with regular subscription option selected, he/she must be logged in as a registered customer to place order with subscriptions. In other case customer will see error message informing that he/she should log in to proceed. 'Checkout as Guest' option is excluded.



The screenshot shows a checkout page with a blue header bar containing the word "Checkout". Below the header, a red arrow points to the text "Log in to place order with subscription". This text is positioned above a login form titled "Returning Customers". The form contains the following elements:

- \* Email: A text input field with a small icon of a document with a plus sign on the right.
- \* Password: A text input field with a small icon of a document with a plus sign on the right.
- ☐ Remember me: A checkbox with the text "Remember me" next to it.
- [forgot password?](#): A link to the forgot password page.
- Login: A blue button with the text "Login".
- [Login with Google](#): A button with the Google logo and the text "Login with Google".
- [Login with Facebook](#): A button with the Facebook logo and the text "Login with Facebook".


## Order Summary Right Rail and Minicart

For products with subscription selected options will be shown on Order Summary Right Rail block and on Minicart.

MiniCart:

Your shopping cart

[View Cart](#)



Color: Green  
In Stock  
Regular Delivery  
Frequency: Monthly

Each	Quantity	Total
<b>\$38.00</b>	<div>1</div>	<del>\$38.00</del> <b>\$30.40</b>

Discount

Estimated Total

\$30.40

Checkout

Right Rail:

Shipping

Shipping To

1 Test - David King 1 Test Baltimore, MD 21224

[Update Address](#) [Add New](#)

Shipping Method

☒ Ground (7-10 Business Days)..... \$5.99

☐ 2-Day Express (2 Business Days)..... \$9.99

☐ Overnight (Next Day)..... \$15.99

☐ This is a Gift

Payment

Next: Payment

Need Help? Call 1-800-555-0199

Order Summary

Subtotal..... \$30.40

Shipping..... \$5.99

Sales Tax..... \$1.82

Total..... \$38.21

1 Items..... \$30.40



Color: Green  
In Stock  
Regular Delivery  
Frequency: Monthly

Each	Quantity	Total
<b>\$38.00</b>	<b>1</b>	<del>\$38.00</del> <b>\$30.40</b>


## Order Confirmation Page

When the order summary page template has been updated, you should be able to see selected subscription options for product in order.

1 Items .....

\$30.40

Multi Color Scarf



Color: Green  
Regular Delivery  
Frequency: Monthly

Quantity	Total
1	<del>\$38.00</del>
	<b>\$30.40</b>

Subtotal .....

\$30.40

Shipping .....

\$5.99

Sales Tax .....

\$1.82

**Total .....**

**\$38.21**

Continue Shopping



## 6. Known Issues

## 7. Release History

Version	Date	Changes
17.1.0	<DATE>	Initial release