

# **CyberSource Storefront Reference Architecture LINK Cartridge Developer Guide**

*Version 20.1.0*



## Table of Contents

1.	Introduction to StoreFront Reference Architecture .....	3
2.	CyberSource Cartridge Overview .....	3
3.	CyberSource SFRA Cartridge Architecture.....	4
4.	Installation Guide.....	5
4.1.	Workspace Preparation .....	5
5.	Cartridge Installation .....	7
6.	Tax Calculation .....	9
7.	Credit Card Authorization.....	10
8.	Delivery Address Verification .....	12
9.	Address Verification Service (AVS) .....	13
10.	Device Fingerprint .....	14
11.	Decision Manager .....	15
12.	Decision Manager Order Update Job.....	16
13.	Payment Tokenization .....	18
14.	Subscription Token Creation .....	19
15.	Apple Pay.....	20
16.	PayPal .....	22
16.1.	PayPal Express.....	23
16.2.	PayPal Credit .....	24
16.3.	PayPal Billing Agreement.....	25
17.	Payer Authentication.....	27
18.	Secure Acceptance Hosted Checkout – iFrame .....	29
19.	Secure Acceptance Redirect .....	31
20.	Secure Acceptance Checkout API .....	33
21.	Secure Acceptance Flex MicroForm .....	35
22.	Capture Service.....	37
23.	Auth Reversal Service.....	39
24.	Credit Service.....	41
25.	Request Customizations.....	42
26.	Visa Checkout.....	43
27.	Bank Transfer.....	45
28.	Alipay.....	48
29.	Google Pay.....	50

## 1. Introduction to StoreFront Reference Architecture

The basic tenants of Salesforce Commerce Cloud's StoreFront Reference Architecture focus on the idea of code separation and modularization. SRFA allows vendors to develop cartridges that extend, append, or replace code in the base storefront cartridge. Applying these ideas individually to views, models, and controllers, we are able to customize any part of the storefront without directly modifying the code in the storefront cartridge. This grants clients the option to pull bug fixes, and feature updates that SFCC periodically provides to the storefront, without the need to re-integrate 3<sup>rd</sup> party cartridges. It also allows 3<sup>rd</sup> party cartridge vendors to deliver updates to their cartridges, that can be quickly and easily integrated by existing customers. As a rule, a cartridge integration cannot require any changes to the base storefront cartridge. As such, you should find the integration of this, and other SFRA cartridges to be less time consuming than you may have experienced on legacy platforms.

## 2. CyberSource Cartridge Overview

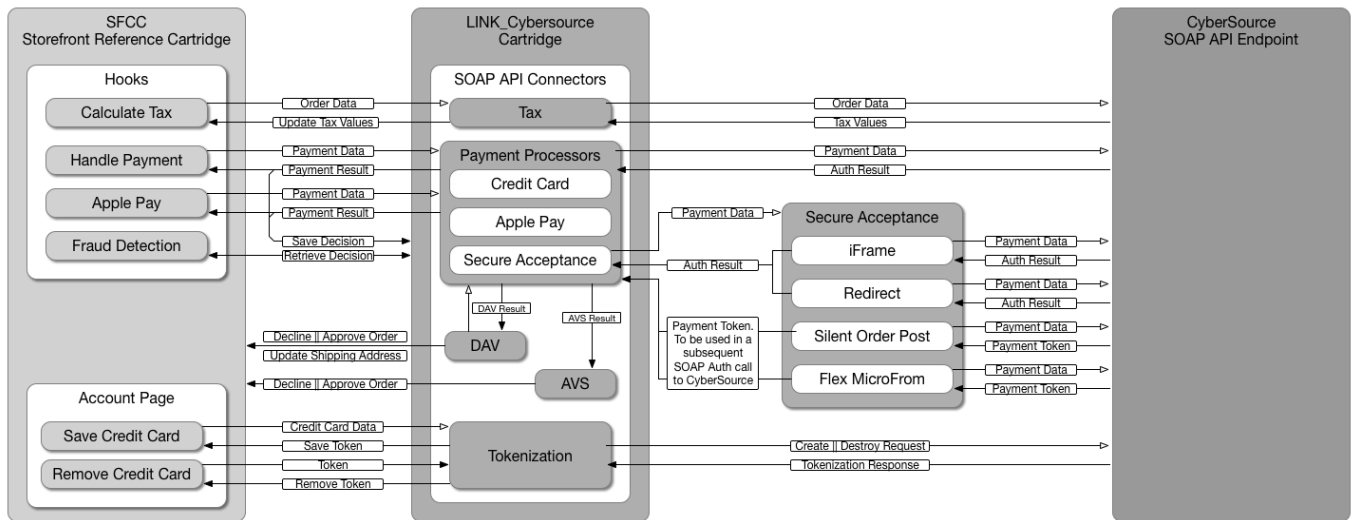
The CyberSource package contains four cartridges. A core cartridge (**int\_cybersource**) that contains core API integrations, including the building and handling of API requests, and parsing responses into objects usable by the storefront. The two legacy architecture cartridges (**int\_cybersource\_pipelines**, **int\_cybersource\_controllers**) each contain sets of wrappers that connect the core code to their respective SFCC platforms. If you are integrating CyberSource with a controller or pipeline version of Site Genesis, please disregard this document, and refer to the corresponding integration guide for your version of Site Genesis. Version 18.1 and higher of the CyberSource cartridge package adds a fourth cartridge (**int\_cybersource\_sfra**), which combines a modified version of the core code that exists in the **int\_cybersource**, along with the necessary hooks, extensions, and wrappers to connect this code to the SFRA storefront. When integrating CyberSource with SFRA, you should only upload the **int\_cybersource\_sfra** cartridge to your workspace and storefront. The remaining three cartridges are not utilized in this integration and can be ignored. The following pages describe the high-level architecture of the SFRA CS Architecture, along with details regarding specific Integrations.

### 2.1 Compatibility

**This version of the Cybersource cartridge is not compatible with versions of SFRA higher than Release 4.3.0. This version can be found on the Master branch of the SFRA repository at commit 06faec3735028e2e52feee840ebfdd8ed684c409 [06faec3] on August 26<sup>st</sup> 2019. This version is compatible with Salesforce B2C Commerce 19.10 release.**

### 3. CyberSource SFRA Cartridge Architecture

## CyberSource SFRA Link Cartridge Overview

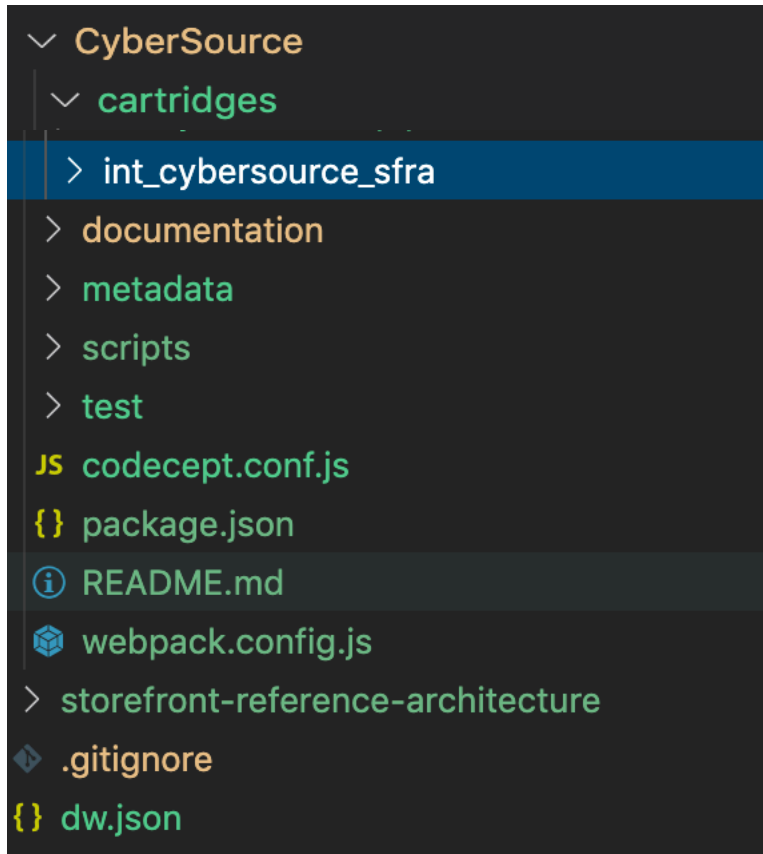


This high-level diagram displays the key connection points and data paths through the various services provided by CyberSource. The LINK cartridge connects many services to the storefront via hooks made available by the SFRA storefront. Some Services, such as AVS, DAV, and Fraud Detection do not utilize their own API requests, and are handled in the Authorization, or Payment Processor API Requests. An overview of each service, along with usage details, is provided below.

## 4. Installation Guide

### 4.1. Workspace Preparation

Create a folder “CyberSource” in your workspace and place the cartridge downloaded from Marketplace. Out of the box, this cartridge (int\_cybersource\_sfra) and storefront-reference-architecture are placed in the directory as shown below. If you have a different project set-up, you will need to open the file ‘/package.json’ and modify the paths.base value to point to your ‘app\_storefront\_base’ cartridge. This path is used by the JS and SCSS build scripts. Once complete, follow the below steps.



1. On your terminal, navigate to CyberSource
2. If you have not already, install node using ‘nvm install node’
3. Run ‘npm install’ to install all of the local dependencies.
4. If using Eclipse, add the ‘int\_cybersource\_sfra’ cartridge to your workspace.
5. If using Visual Studio Code, do the following:
  - a. Create the file ‘dw.json’ under the root folder with contents:

```
{
  "hostname": "your-sandbox-hostname.demandware.net",
  "username": "yourlogin",
  "password": "yourpwd",
  "version": "version_to_upload_to",
  "cartridge": [
    "int_cybersource_sfra",
```

```
    "app_storefront_base",  
    "modules"  
  ]  
}
```

Note: Configuration of this file, and the cartridges you want uploaded and watched will depend on your project.

- b. Run ``npm run uploadCartridge`` command to upload cartridges defined in `dw.json` to the server defined in `dw.json`.
6. Run ``npm run compile:scss`` to compile all `.scss` files into CSS.
7. Run ``npm run compile:js`` - Compiles all `.js` files and aggregates them.
8. If using Eclipse, refresh your project contents, as new JS and css files may have been created, that need to be uploaded. To avoid doing this every time you compile your JS or SCSS, Enable the Workspace Preference ".

Note: If you have trouble getting your build scripts to run, or encounter errors in your npm install, try setting your node version to 8.11.3 and go back to step 3.

## 5. Cartridge Installation

Whether installing the cartridge for the first time, or upgrading to a new version, perform the following steps to ensure you have the latest metadata.

After following the above steps, the `int_cybersource_sfra` cartridge should be uploaded to your SFCC instance, to the active code version. Follow the steps below to configure your server for the `int_cybersource_sfra` cartridge.

1. In Business Manager, navigate to 'Administration > Sites > Manage Sites > Your Site' and select the 'Settings' tab.
2. Add 'int\_cybersource\_sfra:' to the left side of the cartridge path, before 'app\_storefront\_base'

[Administration](#) > [Sites](#) > [Manage Sites](#) > RefArch - Settings

General **Settings** Cache Site Status Page Meta Tag Rules

### RefArch - Settings

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

Instance Type:

ⓘ Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("SEO > Aliases Configuration"). The HTTP/HTTPS hostname values set in this section will be used in the configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges:

Effective Cartridge Path: int\_cybersource\_sfra:app\_storefront\_base;plugin\_apple\_pay;plugin\_facebook;plugin\_pinterest\_commerce;plugin\_web\_payments;bc\_content:core

3. Navigate to 'Administration > Site Development > Import & Export'.
4. Upload meta data zip file /metadata/sfra\_meta.zip and import:
  - [CS SFRA Metadata v20\\_1.xml](#)
  - [CS SFRA CustomObjectDefinitions v20\\_1.xml](#)
5. Navigate to 'Administration > Operations > Import & Export'
6. Upload meta data zip file /metadata/sfra\_meta.zip and import the services and scheduled job data files:
  - [CS SFRA Services v20\\_1.xml](#)
  - [CS SFRA ScheduledJobs v20\\_1.xml](#)
7. Navigate to 'Merchant Tools > Ordering > Import & Export'
8. Upload metadata zip file /metadata/sfra\_meta.zip and import the payment methods data files:
  - [CS SFRA PaymentMethods v20\\_1.xml](#)
9. Navigate to 'Merchant Tools > Site Preferences > CyberSource: Core'
10. Enter the Merchant ID that CyberSource provided you with in the 'CyberSource Merchant ID' field.
11. Generate a SOAP API key in the CyberSource Business Center.
  - Log into CyberSource Business Center.
  - Navigate to 'Account Management > Transaction Security Keys'
  - Select 'Security Keys for the SOAP Toolkit API'
  - Click the 'Generate Key' button.
  - Copy the key at the bottom of the page and save it somewhere secure. You may need it later.

12. Navigate back to the CyberSource: Core Custom Site Preferences.
13. Enter the Merchant Key that you just created in the 'CyberSource Merchant Key' field.
14. Select the appropriate 'CyberSource Endpoint' for your instance. Production should point to the 'Production' CS endpoint, and all other instances should point to the 'test' CS endpoint.
15. Enter your Developer ID, and Merchant ID supplied by CyberSource, in the corresponding fields.
16. Scroll to the top of the page and click 'Save'.
17. Basic cartridge installation is complete. Reference the 'Usage' and 'Configuration' guides on the following pages for activation and configuration of individual services. Some services will require importing additional data files to your sandbox. Instructions to do so are included those specific services 'Usage' sections.



## 6. Tax Calculation

### *Integration Overview*

The CyberSource tax service is integrated via the SFRA OOTB `dw.order.calculateTax` hook. The `calculateTax` hook is registered in the `hooks.json` file with script `./cartridge/scripts/hooks/tax/taxes`. This script acts as a wrapper to the core CyberSource Tax code. Tax values are retrieved from CyberSource and updated in the basket. The hook returns a Status object, preventing the OOTB `calculateTax` hook from being called, thus taking tax calculation priority away from the storefront cartridge. If you intend to use CyberSource as your tax calculator, you should not have another cartridge ahead of this one, that also calls the `calculateTax` hook. In the case of an error, or unresponsive endpoint, the OOTB `calculateTax` script will be used, as a back-up. Product information is provided on an individual line item basis and all merchant/request IDs are captured for future reference. When the customer enters shipping information, the Tax Service is called to calculate taxes. Taxes will only be recalculated when a change has been made to the cart that can affect the total tax amount.

### *Implementation*

To use CyberSource Tax services on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group. Work with CyberSource to ensure tax services are activated and functioning on your account. Follow the steps below to configure the service in Business Manager.

1. Merchandise the `taxCode` field on all products in your catalog.
2. Optionally, you can set the 'CS Tax Calculation Default Product Tax Code' site preference to act as a fallback tax code for all products without a merchandised tax code.
3. Determine and set all associated site preference listed below.
4. Set the 'CS Tax Calculation Enabled' site preference to 'Yes' to enable the service.

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS Tax Calculation Enabled	Enable or disable CyberSource tax service.
CS Tax Calculation Nexus States List	List of states to charge tax in.
CS Tax Calculation No Nexus States List	List of States to not charge tax in.
CS Tax Calculation Default Product Tax Code	Default tax code used when tax code is not set on a product.
CS Tax Calculation Purchase Order Acceptance City	Purchase order acceptance state code.
CS Tax Calculation Purchase Order Acceptance Zip Code	Purchase order acceptance zip code.
CS Tax Calculation Purchase Order Acceptance Country Code	Purchase order acceptance country code.
CS Tax Calculation Purchase Order Origin City	Purchase order origin city.
CS Tax Calculation Purchase Order Origin StateCode	Purchase order origin state code.
CS Tax Calculation Purchase Order Origin ZipCode	Purchase order origin zip code.
CS Tax Calculation Purchase Order Origin Country Code	Purchase order origin country code.
CS Tax Calculation ShipFrom City	Ship from city.
CS Tax Calculation ShipFrom StateCode	Ship from state code.
CS Tax Calculation ShipFrom ZipCode	Ship from zip code.
CS Tax Calculation ShipFrom Country Code	Ship from country code.

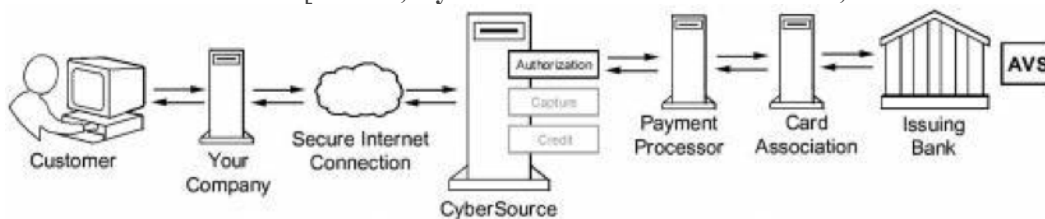
## 7. Credit Card Authorization

### *Integration Overview*

The CC Auth service is integrated via the SFRA OOTB dynamically generated `app.payment.processor.cybersource_credit` hook. The `cybersource_credit` hook is registered in the `hooks.json` file with script `./cartridge/scripts/hooks/payment/processor/cybersource_credit`. This script acts as a wrapper to the core CyberSource Authorization code. Behind this wrapper, an API request is constructed, sent to CS, and the response parsed. In the case of a successful authorization (response code 100), the hook returns a JSON object without an error. All other response codes received result in an error being present in the return object, triggering the storefront to display an error message, and not create the order. Actions taken when making the Authorization call are as follows:

- 1) Creates CyberSource authorization request using ship-to, bill-to, credit card data, and purchase total data from the current basket.
- 2) If authorize Payer is configured, then make the authorize payer request. If not, ignore and continue with the authorization request.
- 3) Create a credit card authorization request.
- 4) If DAV is enabled, set up DAV business rules, as needed.
- 5) Set up AVS if enabled.
- 6) Make the service call to CyberSource via the SOAP API.
- 7) If Delivery Address Verification is enabled, then:
  - a. Capture pertinent DAV result information & DAV Reason Code. Update shipping address if a suggestion was returned and the 'CS DAV Update Shipping Address With DAV Suggestion' site preference is enabled.
  - b. If DAV fails and DAV On Failure is set to 'REJECT', then exit immediately with rejection response
- 8) If DAV On Failure is set to 'APPROVE' and the DAV Reason Code is a fail code (not 100), then:
  - a. Exit immediately with declined or review response, as merchant defines
- 9) Capture pertinent AVS information.
- 10) Capture Fraud response in a session variable to be handled later.
- 11) Validate authorization reason code and set corresponding values, based on Auth response code.

The list of activities depicted in the following diagram takes place when API request is made for an online credit card authorization: [Source, CyberSource Credit Card Service, and October 2009]



1. The customer places an order and provides the credit card number, the card expiration date, and other information about the card.

2. You send a request for authorization over a secure Internet connection. If the customer buys a digitally delivered product or service, you can request both the authorization and the capture at the same time. If the customer buys a physically fulfilled product, do not request the capture until you ship the product.
3. CyberSource validates the order information, and then contacts your payment processor and requests authorization.
4. The processor sends the transaction to the card association, which routes it to the issuing bank for the customer's credit card. Some card companies, including Discover and American Express, act as their own issuing banks.
5. The issuing bank approves or declines the request. Depending on the card type, the bank could also use the Address Verification Service (AVS) to determine whether the customer provided the correct billing address. For more information about AVS, refer to AVS service documents via the CyberSource Services Documentation at [http://www.cybersource.com/support\\_center/support\\_documentation/services\\_documentation/payment.php](http://www.cybersource.com/support_center/support_documentation/services_documentation/payment.php) or as described in this integration guide.
6. CyberSource runs its own tests, and then tells you if the authorization succeeded.
7. Response is sent back to the client.

## *Implementation*

To use CyberSource Tax services on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group and work with CyberSource to ensure tax services are activated and functioning on your account. Follow the steps below to configure the service in Business Manager.

1. Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip into your sandbox.
2. Under 'Merchant Tools > Ordering > Payment Methods' Make sure the 'CREDIT\_CARD' payment method is enabled and configured to use the CYBERSOURCE\_CREDIT payment processor.
3. On the same page, select 'Credit/Debit cards' to enable the credit card types you want supported.

## *Configuration*

Other than the standard requirements of all CyberSource services covered in the installation guide, there are no site preferences associated with a basic Authorization call. But, many services utilize the Credit Card Authorization API to communicate with CyberSource. Please see the sections of any of these services you may be using, to determine how to configure them. i.e. DAV, AVS, Decision Manager.

## 8. Delivery Address Verification

### *Integration Overview*

The Delivery Address Verification does not make an independent API call to validate address data. The shipping address is attached to the payment processing API call and verified in the same request. This means, in its current state, the service cannot be used as a stand-alone address validator on shipping address submission. The validation does not happen until the payment processor is called, which is after the customer has confirmed their order details. At the time of the authorization, a payment processing API call to CyberSource will attempt to validate the address. If the address is valid, a '100' response code is returned, and the order is processed by the storefront as usual. If CyberSource was able to correct mistakes in the supplied address, this corrected address will be sent back in the response. If the appropriate site preference is selected, this address will be automatically copied into the order shipping address, overriding the customer's entry. If a corrected address could not be determined, an error will be passed back to the storefront, and the order is either created or not, based on a site preference value.

### *Implementation*

To use the DAV service, ensure you have followed all steps in the "Cartridge Installation" guide above. As this service runs through the Authorization API, a CyberSource Merchant Id and CyberSource Merchant Key are required. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to ensure the DAV service is activated and functioning on your account. To enable this service, follow the below steps:

1. In the site preference group 'CyberSource: Core', set the 'CS DAV Delivery Address Verification Enabled' preference to 'Yes'
2. Configure the site preferences listed below based on your business needs.

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS DAV Delivery Address Verification Enabled	Enable or disable the DAV service.
CS DAV Update Shipping Address With DAV Suggestion	Update the shipping address with the CS suggestion, if found.
CS DAV On Failure	Accept or Reject the order if DAV fails.

## 9. Address Verification Service (AVS)

### *Integration Overview*

AVS does not yet exist as a stand-alone callable service. The service is performed during an Authorization request. Please refer to the Credit Card Authorization Service overview to understand how AVS has been integrated. A standalone Delivery Address Verification service, with SFRA compatible UI is slated for release with version 19.2.0 of this cartridge.

### *Implementation*

Assuming you have implemented the Credit Card Authorization service, you are ready to use the AVS service. Configure the below site preferences to suite your business needs.

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS AVS Ignore AVS Result	Effectively enables or disables the AVS service.
CS AVS Decline Flags	Leave empty to follow CS default decline flag strategy. Enter flags separated by commas to overwrite the default flag rules.

## 10. Device Fingerprint

### *Integration Overview*

Device Fingerprint collection is handled in the `htmlhead.isml` template. An include in the `<head>` of every page calls the `CYBDeviceFingerprint-GetFingerprint` endpoint. This controller uses a session variable to remember the last time a device fingerprint was generated for this session. If one has not been created in a time greater than a given site preference value, a flag is sent to the rendering template to generate a new one. The session variable is then updated with the current time. In this way, every user will have a device fingerprint generated on their first visit, and the device fingerprint for the user will be updated every N milliseconds. By default, device fingerprints expire every 24 hours, and a customer session can be longer than that, requiring this periodic regeneration.

The fingerprint is generated through a JS include in the `<head>` element, which pulls self-executing code from a third-party service. This data is stored as a digital fingerprint, along with the storefront session ID of the user on the 3<sup>rd</sup> party services servers. CyberSource is able to look up the fingerprint using the session ID that is passed in an Authorization request. The fingerprint can then be used in CyberSource Decision Manager rules during the Authorization.

### *Implementation*

To use Device Fingerprinting, ensure you have followed all steps in the "Cartridge Installation" guide above. You will need a ThreatMetix URL to retrieve the JS that runs on the site, and an Organization ID. Work with CyberSource to obtain these values. Configure the below site preferences to activate the service.

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS Device Fingerprint Enabled	Enable or Disable the Device Fingerprint Service.
CS Device Fingerprint Organization ID	Device Fingerprint Organization ID
CS Device Fingerprint ThreatMetrix URL	URL pointing to JS that generates and retrieves the fingerprint.
CS Device Fingerprint Time To Live	Time, in milliseconds between generating a new fingerprint for any given customer session.

## 11. Decision Manager

### *Integration Overview*

The Decision Manager integration does not make an independent API call to retrieve a fraud decision. A flag is passed through the various payment method API requests, indicating to CyberSource whether or not to run DM rules against the transaction. The results of the decision will be passed back in the API response, and saved in "session.custom.CybersourceFraudDecision" for later use. SFRA utilizes a separate hook, after payment processing, that handles fraud detection. This cartridge makes use of this hook, by subscribing to "app.fraud.detection" with script './cartridge/scripts/hooks/fraudDetection'. The fraudDetection script reads the stored value in the CybersourceFraudDecision session variable and returns the expected value back to SFRA. In the case of an ACCEPT, the hook will return a status of 'success'. In the case of a 'REVIEW' decision, the hook will return 'flag'. In the case of a 'REJECT' decision, the hook will return 'fail'. The SFRA storefront cartridge handles these responses as such:

'success' will allow the order to be created, and Order confirmation status is set to 'Confirmed'.

'flag' will allow the order to be created, and Order confirmation status is set to 'Not Confirmed'.

'fail' will prevent the order from being placed and display a general error message to the customer.

### *Implementation*

To use Decision Manager, ensure you have followed all steps in the "Cartridge Installation" guide above. As this service runs through the Authorization API, a CyberSource Merchant Id and CyberSource Merchant Key are required. Enter these values in the corresponding site preferences under the "CyberSource Core" group and work with CyberSource to ensure the Decision Manager is activated and functioning on your account. To enable this service, work with CyberSource to configure your fraud rules in CS Business Center. All Decision Manager configurations are done through the CS portal, not in your storefront Business Center.

Once your Fraud rules have been configured, set the below site preference to 'Yes' to enable the feature on your storefront.

SFRA storefront versions 3.2.0 or lower contain a hook that interfere with this service. While the hook manager has been updated in later versions of SFRA to prevent this, the CS cartridge is not yet compatible with those storefront versions. As suggested by SFCC, manual removal of the following hook from SFRA is required for this integration to function properly.

Remove

```
{
  "name": "app.fraud.detection",
  "script": "./cartridge/scripts/hooks/fraudDetection"
}
```

From app\_storefront\_base/hooks.json

**Note:** Be sure to read the next section regarding the scheduled job associated with Decision Manager.

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS Decision Manager Enabled	Enable or Disable Decision Manager

## 12. Decision Manager Order Update Job

### *Integration Overview*

This job uses a simple API to retrieve order decisions from CyberSource and update the order confirmation status in SFCC.

As described in the previous section, when Decision Manager is enabled, a certain number of orders will be flagged for review, and not fully confirmed in your SFCC storefront. When integrating with your OMS, you must decide if you want to export orders that are set to 'Not Confirmed'. If you only send Confirmed orders to your OMS, you will need this job to update the confirmation status of the orders that have been reviewed and accepted.

### *Implementation*

If you are not using Decision Manager, you do not need this job. If your OMS asks you to send them 'Not Confirmed' Orders, you may or may not want this job. You will need to determine if the Order confirmation status is required, or desired in SFCC for your business needs.

To Integrate this job into your site, follow the below steps:

1. Navigate to 'Administration > Operations > Job Schedules'
2. Select the Job 'CyberSource: Decision Manager Order Update'
3. Select the 'Step Configurator' tab.
4. Select the Sites you want the Job to run on, from the 'Scope' button.
5. Navigate to the 'Schedule and History' tab and configure the frequency you would like the job to run.
6. Ensure the 'Enabled' check box is selected.

When moving to a production environment, the URL for the API call needs to be updated. This can be done in: Administration > Operations > Services > Service Credentials > ConversionDetailReport – Details

This job is configured based on various site preferences described below:

### *Configuration*

Configure values for below **job parameters** in step **UpdateOrderStatus**

Parameter Name	Usage
MerchantId	CS Merchant ID for the account to get Decisions from.
SAFlexKeyID	Key ID. Work with CS to generate this value.
SAFlexSharedSecret	Shared secret. Work with CS to generate this value.



### Job Parameters: UpdateOrderStatus

Preference Name	Site Pref Group	Usage
CS Decision Manager OrderUpdate Lookback time	Core	Number of hours the job will look back for new decisions. CS does not support lookbacks over 24 hours. Do not set above 24.
Secure Acceptance Flex Host Name	Secure Acceptance	Host Name. CS can provide this value.

## 13. Payment Tokenization

### *Integration Overview*

Tokenization is the replacement of sensitive data with a unique identifier that cannot be mathematically reversed. When this service is enabled, a token will be generated when a customer saves a credit card. This token is used when making a payment with that saved card. Typically, the token will retain the last four digits of the card as a means of accurately matching the token to the payment card owner. The remaining numbers are generated using proprietary tokenization algorithms.

Tokenization is implemented in two places:

#### **The Billing Step**

During checkout, a registered and logged in user has the option to save the credit card they are using to make the purchase. When they choose this option, and confirm the order, the route 'CheckoutServices- PlaceOrder' is being replaced by a new version in the CyberSource cartridge. This script contains all of the code present in the SFRA storefront version at time of development, with additions that make an API call to CS requesting a token be generated before the Authorization is made. CyberSource will respond with a token, which is saved to the payment, and sent in the subsequent Authorization call.

#### **The Account: Payment Instruments Page**

Customers also have the option of creating and deleting payment methods from their Account Page: Payment Instrument. To achieve tokenization from here, the 'PaymentInstruments-SavePayment route was replaced. All of the SFRA code has been copied into this script, along with an addition of making an API call to request a token and saving the token to the Payment Instrument.

Deleting a Card works similarly. The 'PaymentInstruments-DeletePayment route has been replaced and modified to send an API request to CyberSource to delete the token, before deleting the Payment Instrument in SFCC.

### *Implementation*

To use Tokenization, ensure you have followed all steps in the "Cartridge Installation" guide above. Work with CyberSource to ensure the Tokenization is activate and working on your account. Configure the below site preferences to activate the service on your storefront:

### *Configuration*

#### **Site Preference Group: CyberSource: Core**

<b>Preference Name</b>	<b>Usage</b>
CS Tokenization Enabled	Enable or Disable the Tokenization Service.

## 14. Subscription Token Creation

### *Integration Overview*

When making an authorization call to CyberSource, you have the option to request a subscription token be created. When enabled, your authorization calls will include a request to generate this token. The token will be returned in the response object and saved to the payment instruments 'creditCardToken' field. You can then utilize this token in other third-party integrations. For example, you can send it to your OMS, to allow them to perform Future Authorizations, Captures, Reversals, etc.

### *Implementation*

To enable subscription token generation, ensure you have followed all steps in the "Cartridge Installation" guide above. Configure the below site preference to enable the service:

### *Configuration*

#### Site Preference Group: CyberSource: Core

Preference Name	Usage
CS Subscription Tokens Enabled	Enable or Disable the option to generate subscription tokens.

## 15. Apple Pay

### *Integration Overview*

The CyberSource apple pay authorization service is integrated via the SFRA OOTB “dw.extensions.applepay.paymentAuthorized.authorizeOrderPayment”. The apple pay authorizeOrderPayment hook is registered in the hooks.json file with script “/cartridge/scripts/hooks/applepay/applePayAuth”.

This script acts as a wrapper to the core CyberSource apple pay authorization code. Apple Pay payment card data is sent to CyberSource for authorization and authorization status is retrieved from CyberSource. The hook returns a Status object, preventing the OOTB apple pay authorizeOrderPayment hook from being called, thus taking apple pay authorization priority away from the storefront cartridge.

### *Implementation*

#### Business Manager Configuration

1. Go to: “Merchant Tools > Site Preferences > Apple pay
2. Check “Apple Pay Enabled?”
3. Fill in the “Onboarding” form:
  - a. Ensure “Apple Merchant ID” and “Apple Merchant Name” match settings in your apple account
  - b. Ensure all other fields match the your supported CyberSource settings
4. Fill in the “Storefront Injection” form:
  - a. Selects where apple pay buttons should be displayed on your site.
5. Fill in “Payment Integration” form:
  - a. Leave all form fields blank
  - b. Ensure “Use Basic Authorization” is unchecked

#### CyberSource Business Center Configuration

1. Go to: “Account Management > Digital Payment Solution
2. Click “Sign Up” next to Apple Pay
3. Follow the step by step instructions on the page

### *Configuration*

The CyberSource Apple Pay integration does not make use of any custom site preferences. See the following SFCC Apple Pay preferences for configuration:

### Site Preferences: Apple Pay

Preference Name	Usage
Apple Pay Enabled?	Enable or Disable the Apple Pay Service.
Apple Merchant ID	Match settings in your Apple account.
Apple Merchant Name	Match settings in your Apple account.
Country Code	Match setting in your CyberSource account.
Merchant Capabilities	Match setting in your CyberSource account.
Supported Networks	Match setting in your CyberSource account.
Required Shipping Address Fields	Match setting in your CyberSource account.
Required Billing Address Fields	Match setting in your CyberSource account.
Inject Apple Pay Button on Mini Cart?	Display Apple Pay button in the mini cat.
Inject Apple Pay Button on Cart Page?	Display Apple Pay button in the cart.
Redirect Pages to HTTPS?	Redirect Pages to HTTPS
Use Commerce Cloud Apple Pay Payment API?	Not used. Leave blank.
Payment Provider URL	Not used. Leave blank.
Payment Provider Merchant ID	Not used. Leave blank.
API Version	Set to v1
Use Basic Authorization	This must be enabled.
Payment Provider User	Not used. Leave blank.
Payment Provider Password	Not used. Leave blank.
Use JWS?	Not used. Leave blank.
JWS Private Key Alias	Not used. Leave blank.

## 16. PayPal

### *Generic PayPal Configuration*

Prior to development phase, there are a generic set of configurations that a development team needs to account for. These configurations include:

1. PayPal developer account
2. PayPal sandbox account
3. Linking developer and sandbox account. On creating a PayPal developer account, get in touch with the CyberSource team, share the developer account details and get the developers' details configured on CyberSource (BackOffice Configuration tool).

Screenshot of the detailed set of configurations for #1 & #2

#### Step-by-step guide

1. Register for a PayPal developer account.
2. Login at <https://developer.paypal.com> with your "normal" developer email (ex. [mitaylor@cybersource.com](mailto:mitaylor@cybersource.com)).
3. Click on "Accounts" under "Sandbox" to ensure you have at least one of both business and personal sandbox accounts.
  - a. Format is [email-buyer@domain.com](mailto:email-buyer@domain.com) and [email-facilitator@domain.com](mailto:email-facilitator@domain.com)
  - b. Ex. [mitaylor-buyer@cybersource.com](mailto:mitaylor-buyer@cybersource.com) (sandbox consumer account) and [mitaylor-facilitator@cybersource.com](mailto:mitaylor-facilitator@cybersource.com) (sandbox business account).
  - c. If you don't have those, create them using the "Create Account" link in the top right corner of the page:

### Sandbox Accounts

[Create Account](#)

Questions? Check out the [Testing Guide](#). Non-US developers should read our [FAQ](#).

To link your sandbox account to your developer account, [log in with PayPal](#) and provide your sandbox account credentials.

Total records: 3

<input type="checkbox"/>	Email Address	Type	Country	Date Created	
<input type="checkbox"/>	› <a href="mailto:mitaylor-buyer@cybersource.com">mitaylor-buyer@cybersource.com</a>	PERSONAL	US	03 May 2017	
<input type="checkbox"/>	› <a href="mailto:mitaylor_buyer@cybersource.com">mitaylor_buyer@cybersource.com</a>	PERSONAL	US	26 Jul 2013	
<input type="checkbox"/>	› <a href="mailto:mitaylor-facilitator@cybersource.com">mitaylor-facilitator@cybersource.com</a>	BUSINESS Pro	US	26 Jul 2013	

[Delete](#)

4. Click "My Apps & Credentials" under "Dashboard".
5. Scroll to the "REST API Apps" section.
6. Click "Create App".
7. Give it whatever name you want, making sure to link it to your business account:

**Application Details**

**App Name**

On successful completion of #1 & #2, share the following keys with Cybersource:

ClientID  
Secret  
Merchant Account ID  
Merchant email

Document on where to find this information can be found here:



Merchant PayPal  
Account Info.docx

## 16.1. PayPal Express

### *Integration Overview*

PayPal Express provides a set of services which enables you to checkout in a faster and safer manner. PayPal integration with CyberSource provides 3 ways to complete the checkout.

1. Minicart
2. Cart Page
3. Billing Page

The CyberSource cartridge provides in-context checkout option i.e. when a customer clicks on Checkout with PayPal on the checkout page or mini cart, the website remains in view while a PayPal window appears. The Customer logs in and selects a payment method and shipping address and confirms the payment. PayPal will then redirect the customer to your order review page for completion. The CyberSource cartridge enables merchant to select the order type from BM i.e. Custom or Standard.

### *Custom Order*

PayPal Custom Order enables you to perform multiple authorizations and multiple captures for each authorization. Below are the service requests for custom orders:

- **Sessions Service**- Creates a payment with PayPal to set up an order
- **Check Status Service**- Requires the request ID value that was returned in Sessions Service and returns customer information
- **Order Service**- Requires the request ID value that was returned in Sessions Service and Payer ID, creates the order in anticipation of one or more authorization
- **Authorization Service**- Requires request ID value that was returned in the order response, obtains the authorization
- **Capture Service**- Requires the request ID value that was returned in the authorization response and enables you to capture the entire authorized amount

## Standard Order

PayPal Standard Order enables merchants to perform authorize and capture actions at the same time. Below are the service requests for Standard order

- **Sessions Service**- Creates a payment or Billing agreement with PayPal to set up an order
- **Check Status Service**- Requires the request ID value that was returned in Sessions Service and returns customer information
- **Order Service**- Requires the request ID value that was returned in Sessions Service and Payer ID, creates the order in anticipation of one or more authorization
- **Sale Service**- Requires the request ID value that was returned in order response, this service obtains authorization, and captures the authorized amount

## Implementation

To use PayPal Express, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group and work with CyberSource to ensure PayPal services are activated and functioning on your account. Follow the steps below to configure the service in Business Manager

1. Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip into your sandbox
2. Under 'Merchant Tools > Ordering > Payment Methods' Make sure the 'PAYPAL' payment method is enabled and configured to use the 'PAYPAL\_EXPRESS' payment processor

## Configuration

### Site Preference Group: CyberSource\_paypal

Preference Name	Usage
CsEnableExpressPaypal	Effectively enables or disables the PayPal Express checkout.
Paypal Order Type	The type of authorization to follow for PayPal orders. Select STANDARD for Authorize & Capture or select CUSTOM for just Authorize

## 16.2. PayPal Credit

### Integration Overview

The PayPal credit button on your checkout page enables you to offer customer's PayPal Credit as a standalone payment option. While PayPal Express allows a user to select PayPal Credit as a payment method, the PayPal Credit implementation offers a direct connection to the PayPal Credit page, if the custom is not yet enrolled. PayPal Credit leverages the PayPal Express implementation. For PayPal Credit an additional flag 'paymentOptionID', set to true, is included in the Sessions service request. Below are the service requests for PayPal Credit.

- **Sessions Service** with additional flag 'paymentOptionID'
- After getting the payment Transaction ID and request ID from sessions response, **Check Status Service, Order Service & Sale Service** follow as explained above for PayPal express



## Implementation

To use PayPal Credit, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group and work with CyberSource to ensure PayPal services are activated and functioning on your account. Follow the steps below to configure the service in Business Manager

1. Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip into your sandbox Under 'Merchant Tools > Ordering > Payment Methods' Make sure the 'PAYPAL\_CREDIT' payment method is enabled and configured to use the 'PAYPAL\_CREDIT' payment processor

## Configuration

### Site Preference Group: CyberSource\_paypal

Preference Name	Usage
Paypal Order Type	The type of authorization to follow for PayPal orders. Select STANDARD for Authorize & Capture or select CUSTOM for just Authorize

## 16.3. PayPal Billing Agreement

### Integration Overview

A PayPal Express Checkout billing agreement enables you to use Billing agreement ID for billing without requiring customer to specifically authorize each payment. Once the agreement created for customer, customer's Billing agreement ID would be used to Authorize the order. PayPal Billing agreement is applicable only for logged user, when customer checks Billing agreement checkbox from Billing page additional flag billingAgreementIndicator need to include in Session service request. Request ID returned in session service will be used in PayPal Billing agreement service, Billing Agreement ID would be saved in customer profile, this billing agreement ID would be used in further transaction. Cybersource Cartridge allows merchants to enable/disable billing agreement from BM site preferences. Below are the service requests for Billing Agreement

- **Sessions Service** – Creates Billing agreement with PayPal to setup an order
- **Billing Agreement Service**- if customer profile does not contain Billing Agreement ID, this service would create the Billing agreement and saves the Billing agreement ID in customer profile. It requires the request ID value returned in sessions response
- **Check Status Service**- If customer profile contains billing agreement ID , sessions service would be skipped , billing agreement ID would be used in Check Status service
- **Sale Service** – Requires billing agreement ID returned in billing agreement service response. This service obtains authorization, and captures the authorized amount

## Implementation

To use PayPal Billing Agreement, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group and work with CyberSource to ensure PayPal services are activated and functioning on your account. The necessary configurations are already covered in PayPal Express implementation section.

## Configuration

### Site Preference Group: CyberSource\_paypal

Preference Name	Usage
Billing Agreement	Effectively enables or disables the PayPal Billing Agreement.
Paypal Order Type	The type of authorization to follow for PayPal orders. Select STANDARD for Authorize & Capture or select CUSTOM for just Authorize

## 17. Payer Authentication

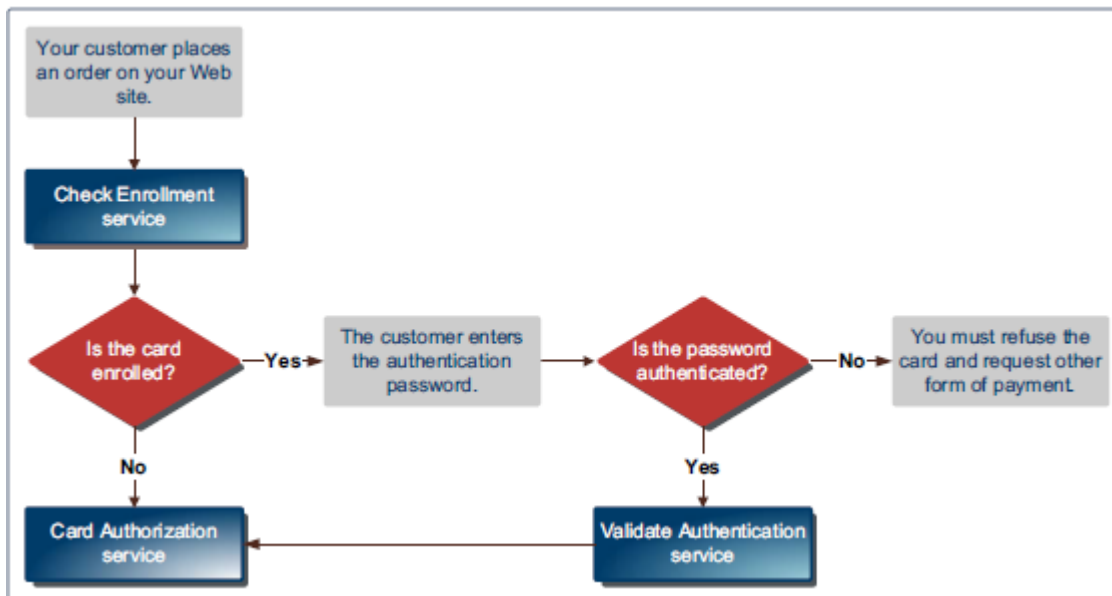
### *Integration Overview*

CyberSource Payer Authentication services enable you to add support to your web store for card authentication services, including Visa Verified by VisaSM, MasterCard® and Maestro® SecureCode™ (UK Domestic and international), American Express SafeKeySM, and JCB J/Secure™. These card authentication services deter unauthorized card use and protect you from fraudulent chargeback activity referred to as liability shift.

#### How It Works

Payer Authentication provides the following services:

1. Check Enrollment: Determines whether the customer is enrolled in one of the card authentication programs.
2. Validate Authentication: Ensures that the authentication that you receive from the issuing bank is valid.



The Check Enrollment service determines whether the customer is enrolled in one of the Card authentication services:

No: If the card is not enrolled, you can process the authorization immediately.

Yes: If the card is enrolled, the customer's browser displays a window where the customer can enter the password associated with the card. This is how the customer authenticates their card with the issuing bank. If the password matches the password stored by the bank, you need to verify that the information is valid with the Validate Authentication service. If the identity of the sender is verified, you can process the payment with the Card Authorization service.

If the password does not match the password stored by the bank, the customer may be fraudulent. You must refuse the card and can request another form of payment.

## Implementation

To enable Payer Authentication services on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, CyberSource Merchant Key and Cybersource PA Merchant ID are required for Cybersource PayerAuthentication Enrollment and PayerAuthentication Validate. Along with these, the cruise credentials ApiIdentifier, ApiKey, OrgUnitID and Merchant Name are needed. Enter these values in the corresponding site preferences under the "CyberSource: Core" group and work with CyberSource to ensure required Credit card types are configured on Cybsersource portal to participate in the Payer Authentication service on your account. Also, work with Cybersource to understand if the ByPass rules are set for your account. ByPass rules define the window of order transaction amounts which then decide whether the Authentication window is presented or bypassed. Sample of Bypass rules:

- 1) Bypass authentication if transaction amount is > \$500 and
- 2) Bypass authentication if transaction amount is < \$100

Import Payment methods – Already covered in section [Credit Card Authorization](#), this section is included for reference

1. Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip into your sandbox.
2. Under 'Merchant Tools > Ordering > Payment Methods' Make sure the 'CREDIT\_CARD' payment method is enabled and configured to use the CYBERSOURCE\_CREDIT payment processor.
3. On the same page, select 'Credit/Debit cards' and check the payer authentication checkbox on any credit card types you want to support Payer Authentication.

## Configuration

Under 'Merchant Tools > Ordering > Payment Methods, select 'Credit/Debit cards'. Select each credit card as needed and ensure custom-attribute 'Enable Payer Authentication' checkbox is checked.

### Site Preference Group: CyberSource Core

Preference Name	Usage
CS PA Merchant ID	Payer Auth merchant ID
CS PA Save Proof.xml	To enable/disable saving of proof.xml in order object
CS PA Save ParesStatus	Default False Save ParesStatus received as response from Pa Authenticate request and send it as param in ccAuth request call. This field should be enabled after verifying cybersource merchant account settings
CruiseApiKey	A shared secret value between the merchant and Cardinal. This value should never be exposed to the public.
CruiseApiIdentifier	GUID used to identify the specific API Key
CruiseMerchantName	Merchant Name
CruiseOrgUnitId	GUID to identify the merchant organization within Cardinal systems
CardinalCruiseApiPath	Songbird.js script API path

## Upgrade to 3DS2.0

If you are currently using CYBS cartridge and would like to upgrade to 3DS2.0, please refer below doc.



3DS2.x Dev Guide  
for SFRA.docx

## 18. Secure Acceptance Hosted Checkout – iFrame

### *Integration Overview*

Secure Acceptance payment gateway is used to process transaction requests directly from the customers browser so that sensitive payment data does not pass through the SFCC servers.

*Secure Acceptance iFrame:* Customer will be redirected to a Secure Acceptance payment gateway within an iFrame, embedded in a new summary page added to the checkout flow.

Secure Acceptance Web/Mobile Authorization Sequence flow:

1. 'SecureAcceptanceAuthorize' function create Request and signature using signed and unsigned field names and validate the request.
2. After successful validation of the request using signature and then ajax call is made show the to complete the checkout flow
3. After successful checkout completion, Customer is returned to Demandware custom controller method.
4. Secure Acceptance response method get the response in CurrentHttpParameterMap, again signature is created using the response data and matched with the response signature, once validated response is parsed
5. Based on Decision and reason code Order will get placed or failed in Demandware.

### *Implementation*

To use CyberSource Secure Acceptance iFrame service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to Secure Acceptance services are activated and functioning on your account. To complete the checkout process successfully create Secure Acceptance profile on CyberSource business center console under ' Tools & Settings > Secure Acceptance > Profiles > Create New Profile '. While creating the profile check the checkbox ' Web/Mobile' for Integration Methods in Profile Information and configure all the mandatory settings and ensure you activate the profile. Follow the steps below to configure the service in Business Manager.

1. Determine and set all associated site preference listed below.
2. Set the 'CsSAType' site preference to 'SA\_IFRAME' to enable the service.

## Configuration

### Site Preference Group: CyberSource Secure Acceptance

Preference Name	Usage
CsSAType	Secure Acceptance Type
SA_Iframe_ProfileID	Secure Acceptance Iframe Profile ID
SA_Iframe_SecretKey	Secure Acceptance Iframe secret key
SA_Iframe_AccessKey	Secure Acceptance Iframe Access Key
CsSAIframeFormAction	CyberSource secure acceptance Iframe form action
CsSAOverrideBillingAddress	CyberSource Secure Acceptance Override Billing Address
CsSAOverrideShippingAddress	CyberSource Secure Acceptance Override Shipping Address
CsCvnDeclineFlags	CyberSource Ignore CVN Result (CVN)

## 19. Secure Acceptance Redirect

### *Integration Overview*

Secure Acceptance payment gateway is used to process transaction requests directly from the customers browser, so that sensitive payment data does not pass through SFCC servers.

*Secure Acceptance Redirect:* Customer will be redirected to a Secure Acceptance payment gateway when clicking on Place Order from Review Page

Secure Acceptance Web/Mobile Authorization Sequence flow:

1. 'SecureAcceptanceAuthorize' function create Request and signature using signed and unsigned field names and validate the request.
2. After successful validation of the request using signature and then its redirected to the secure payment page to complete the checkout flow
3. After successful checkout completion, Customer is returned to Demandware custom controller method.
4. Secure Acceptance response method get the response in CurrentHttpParameterMap, again signature is created using the response data and matched with the response signature, once validated response is parsed
5. Based on Decision and reason code Order will get placed or failed in Demandware

### *Implementation*

To use CyberSource Secure Acceptance Redirect service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to Secure Acceptance services are activated and functioning on your account. To complete the checkout process successfully create Secure Acceptance profile on CyberSource business center console under ' Tools & Settings > Secure Acceptance > Profiles > Create New Profile '. While creating the profile check the checkbox ' Web/Mobile' for Integration Methods in Profile Information and configure all the mandatory settings and ensure you activate the profile. Follow the steps below to configure the service in Business Manager.

1. Determine and set all associated site preference listed below.
2. Set the 'CsSAType' site preference to 'SA\_REDIRECT' to enable the service.

## Configuration

### Site Preference Group: CyberSource Secure Acceptance

Preference Name	Usage
CsSAType	Secure Acceptance Type
SA_Redirect_ProfileID	Secure Acceptance Redirect Profile ID.
SA_Redirect_SecretKey	Secure Acceptance Redirect Secret Key.
SA_Redirect_AccessKey	Secure Acceptance Redirect Access Key.
CsSARedirectFormAction	CyberSource secure acceptance redirect form action.
CsSAOverrideBillingAddress	CyberSource Secure Acceptance Override Billing Address.
CsSAOverrideShippingAddress	CyberSource Secure Acceptance Override Shipping Address.
CsCvnDeclineFlags	CyberSource Ignore CVN Result (CVN).



## 20. Secure Acceptance Checkout API

### *Integration Overview*

Secure Acceptance Checkout API (Silent Order POST) payment gateway is used to process transaction requests directly from the customers browser, so that sensitive payment data does not pass through SFCC servers.

Secure Acceptance Silent Order Post: Credit Card form data is posted to a Secure Acceptance silent post URL where a token is generated and returned to SFCC. The user is redirected to the order review page and standard card authorization flow continues, using the token received from SA.

Secure Acceptance Silent Order Post Authorization Sequence flow:

1. 'SecureAcceptanceAuthorize' function create Request and signature using signed and unsigned field names and validate the request.
2. After successful validation of the request using signature and then ajax call is made show the to complete the checkout flow
3. After successful checkout completion, Customer is returned to Demandware custom controller method.
4. Secure Acceptance response method get the response in CurrentHttpParameterMap, again signature is created using the response data and matched with the response signature, once validated response is parsed
5. Based on Decision and reason code Order will get placed or failed in Demandware.

### *Implementation*

To use CyberSource Secure Acceptance Silent Post service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to Secure Acceptance services are activated and functioning on your account. To complete the checkout process successfully create Secure Acceptance profile on CyberSource business center console under ' Tools & Settings > Secure Acceptance > Profiles > Create New Profile '. While creating the profile check the checkbox ' Silent Order Post' for Integration Methods in Profile Information and configure all the mandatory settings and ensure you activate the profile. Follow the steps below to configure the service in Business Manager.

1. Determine and set all associated site preference listed below.
2. Set the 'CsSAType' site preference to 'SA\_SILENTPOST' to enable the service.

## Configuration

### Site Preference Group: CyberSource Secure Acceptance

Preference Name	Usage
CsSAType	Secure Acceptance Type
SA_Silent_ProfileID	Secure Acceptance Silent Post Profile ID.
SA_Silent_SecretKey	Secure Acceptance Silent Post Secret Key.
SA_Silent_AccessKey	Secure Acceptance Silent Post Access Key.
Secure_Acceptance_Token_Create_Endpoint	Secure_Acceptance_Token_Create_Endpoint.
Secure_Acceptance_Token_Update_Endpoint	Secure_Acceptance_Token_Update_Endpoint.
CsSAOverrideBillingAddress	CyberSource Secure Acceptance Override Billing Address.
CsSAOverrideShippingAddress	CyberSource Secure Acceptance Override Shipping Address.
CsCvnDeclineFlags	CyberSource Ignore CVN Result (CVN).

## 21. Secure Acceptance Flex MicroForm

### *Integration Overview*

Flex Microform provides the most secure method for tokenizing card data. Flex Microform is designed to be simple to integrate and allows you to focus on creating the best possible checkout experience for your customers. The provided JavaScript library enables you to replace the sensitive card number input field with a secure iFrame, hosted by CyberSource, that will capture data on your behalf. This embedded field will look and feel just like any other input in your checkout process allowing you to create a frictionless experience. Once captured, the card number is replaced with a mathematically irreversible token that can only be used by you. The token can be used in place of the card number for follow on transactions in existing CyberSource APIs.

A Flex Microform consists of 2 main components:

1. A server-side component that requests limited use public keys from the Flex API
2. Using the Flex Microform client-side js library to seamlessly replace the sensitive PAN field in your input form.
3. The token created is used to do Credit Card Auth.

### *Implementation*

To use CyberSource Flex Microform service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group. Also, the flex API requires a date in a specific format, and in the English language. The 'en\_US' local must be present in your administration's locals, but does not need to be enabled. The proper date format is configured in the 'en\_US' local's regional setting. Follow the steps below to configure the service in Business Manager.

1. To complete the checkout process successfully, create API's Key and shared secret on CyberSource business center console 'Account Management > Client Integration Management > Payment Configuration > Key Management'. Click Generate key and select API Cert/Secret > Shared Secret and Submit. Download or Copy the Shared Secret generated.
2. To get the key id, select Key Management and filter provide API Keys and All Keys.
3. In Business Manager, go to Administration > Customization > Services and click on the 'cybersourceflextoken' Credentials. Ensure the appropriate URL is set for the environment you are configuring.
  1. Test: <https://apitest.cybersource.com/flex/v1/keys>
  2. Production: <https://api.cybersource.com/flex/v1/keys>
4. Determine and set all associated site preference listed below.
5. Set the 'CsSAType' site preference to 'SA\_FLEX' to enable this service.
6. Navigate to 'Administration > Global Preferences > Locales'
7. Ensure the local 'en\_US' is present.

8. If not present, create a new local with the following information:
  - Language Code: [en](#)
  - Country Code: [US](#)
  - XML Code: [en-US](#)
  - ISO-3 Language: [eng](#)
  - ISO-3 Country: [USA](#)
  - Fallback Locale: [English\(en\)](#)
9. Once present, select the 'en\_US' local, and navigate to the 'Regional Settings' tab.
10. In the 'Long Date Pattern' field, enter the sting: [EEE, dd MMM yyyy HH:mm:ss z](#)
11. Apply these changes, and configuration is complete.

## Configuration

### Site Preference Group: CyberSource Secure Acceptance

Preference Name	Usage
CsSAType	Secure Acceptance Type
SA_Flex_HostName	Test: apitest.cybersource.com Production: api.cybersource.com
SA_Flex_KeyID	Flex Microform Key ID
SA_Flex_SharedSecret	Flex Microform Shared Secret

## 22. Capture Service

### *Integration Overview*

This cartridge contains an interface that allows you to connect with the CyberSource Capture Service. There is no storefront connectivity with this interface, but it is available for you to use in your own integrations. CyberSource supports captures for all processors. When you are ready to fulfill a customer's order and transfer funds from the customer's bank to your bank, capture the authorization for that order. When fulfilling only part of a customer's order, do not capture the full amount of the authorization. Capture only the cost of the items that you ship. When you ship the remaining items, request a new authorization, and then capture the new authorization.

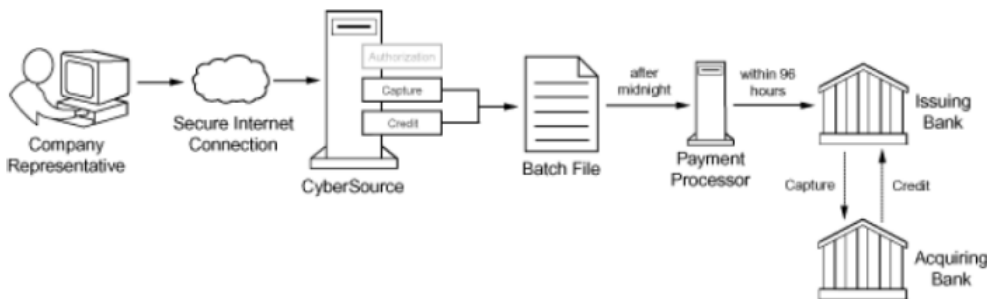
A capture is a follow-on transaction that uses the request ID returned from a previous authorization. The request ID links the capture to the authorization. CyberSource uses the request ID to look up the customer's billing and account information from the original authorization, so you are not required to include those fields in your capture request.

### *Captures*

*Note: This section covers Capture service only for Credit Cards.*

Unlike authorizations, a capture does not happen in real time. All of the capture requests for a day are placed in a batch file and sent to the processor. In most cases, the batch is settled at night. It usually takes two to four days for your acquiring bank to deposit funds in your merchant bank account.

The following figure shows the steps that occur when you request a capture or credit.



1. You send a request for capture or credit over a secure Internet connection.
2. CyberSource validates the order information, then stores the capture or credit request in a batch file.
3. After midnight, CyberSource sends the batch file to your payment processor.
4. The processor settles the capture or credit request and transfers funds to the appropriate bank account.

### *Implementation*

Ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group.

The interface you will use to make capture requests is in the form of a single function:

CCCaptureRequest(requestID, merchantRefCode, paymentType, purchaseTotal, currency)

This function can be found in the script 'scripts/facade/CardFacade.ds'. A working example of how to use this function can be found in the CYBServicesTesting-CaptureService controller. You will first get an instance of the CardFacade object, and make the call as follows:

```
var CardFacade = require('~/cartridge/scripts/facade/CardFacade');  
var serviceResponse = CardFacade.CCCaptureRequest(requestID, merchantRefCode, paymentType, paymentTotal, currency);
```

The resulting serviceResponse object will contain the full response object generated by the request. The contents of this object will determine your logic in handling errors and successes. For detailed explanations of all possible fields and values, reference the Official CyberSource documentation for the CCCapture Service.

### Capture Request Parameters

Parameter Name	Description
requestID	Transaction ID obtained from the initial Authorization
merchantRefCode	SFCC Order Number
paymentType	Payment Type used for the Authorization
purchaseTotal	Order Total
currency	Currency code (ex. 'USD')

## 23. Auth Reversal Service

### *Integration Overview*

This cartridge contains an interface that allows you to connect with the CyberSource Auth Reversal Service. There is no storefront connectivity with this interface, but it is available for you to use in your own integrations as is the case with Credit Card Capture service. The full authorization reversal service releases the hold that the authorization placed on the customer's credit card funds. Use this service to reverse an unnecessary or undesired authorization.

*Note: Each issuing bank has its own rules for deciding whether a full authorization reversal succeeds or fails. When a reversal fails, contact the issuing bank to learn whether it is possible to reverse the authorization by alternate means.*

If your processor supports authorization reversal after void (ARAV), you can reverse an authorization after you void the associated capture. See "Authorization Reversal after Void (ARAV)," page 56 from [http://apps.cybersource.com/library/documentation/dev\\_guides/CC\\_Svcs\\_SO\\_API/Credit\\_Cards\\_SO\\_API.pdf](http://apps.cybersource.com/library/documentation/dev_guides/CC_Svcs_SO_API/Credit_Cards_SO_API.pdf). If your processor does not support ARAV, you can use the full authorization reversal service only for an authorization that has not been captured and settled.

For complete list of supported Processors and Card Types, please refer to *page 49* of:

[http://apps.cybersource.com/library/documentation/dev\\_guides/CC\\_Svcs\\_SO\\_API/Credit\\_Cards\\_SO\\_API.pdf](http://apps.cybersource.com/library/documentation/dev_guides/CC_Svcs_SO_API/Credit_Cards_SO_API.pdf)

### *Implementation*

*Note: This section covers Reversal service only for Credit Cards.*

Ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group.

The interface you will use to make auth reversal requests is in the form of a single function:

`CCAuthReversalService(requestID, merchantRefCode, paymentType, currency, amount)`

This function can be found in the script 'scripts/facade/CardFacade.ds'. A working example of how to use this function can be found in the CYBServicesTesting-CCAuthReversalService controller. You will first get an instance of the CardFacade object, and make the call as follows:

```
var CardFacade = require('~cartridge/scripts/facade/CardFacade');  
var serviceResponse = CardFacade.CCAuthReversalService (requestID, merchantRefCode, paymentType, currency, amount);
```

The resulting serviceResponse object will contain the full response object generated by the request. The contents of this object will determine your logic in handling errors and successes. For detailed explanations of all possible fields and values, reference the Official CyberSource documentation for the CCAuthReversal Service.

### Authorization Reversal Request Parameter

Parameter Name	Description
requestID	Transaction ID obtained from the initial Authorization
merchantRefCode	SFCC Order Number
paymentType	Payment Type used for the Authorization
amount	Order Total
currency	Currency code (ex. 'USD')



## 24. Credit Service

### *Integration Overview*

This cartridge contains an interface that allows you to connect with the CyberSource Credit Service. There is no storefront connectivity with this interface, but it is available for you to use in your own integrations as is the case with Credit Card Capture & reversal services. CyberSource supports credits for all processors. When your request for a credit is successful, the issuing bank for the payment card takes money out of your merchant bank account and returns it to the customer. It usually takes two to four days for your acquiring bank to transfer funds from your merchant bank account. Credit requests are batched in the same manner as captures.

### *Implementation*

*Note: This section covers Credit service only for Credit Cards.*

Ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource: Core" group.

The interface you will use to make credit requests is in the form of a single function:

```
CCCreditRequest(requestID, merchantRefCode, paymentType, purchaseTotal, currency)
```

This function can be found in the script 'scripts/facade/CardFacade.ds'. A working example of how to use this function can be found in the CYBServicesTesting-CreditService controller. You will first get an instance of the CardFacade object, and make the call as follows:

```
var CardFacade = require('~cartridge/scripts/facade/CardFacade');  
var serviceResponse = CardFacade.CCCreditRequest(requestID, merchantRefCode, paymentType, paymentTotal, currency);
```

The resulting serviceResponse object will contain the full response object generated by the request. The contents of this object will determine your logic in handling errors and successes. For detailed explanations of all possible fields and values, reference the Official CyberSource documentation for the CCCredit Service.

#### Credit Request Parameters

Parameter Name	Description
requestID	Transaction ID obtained from the initial Authorization
merchantRefCode	SFCC Order Number
paymentType	Payment Type used for the Authorization
purchaseTotal	Order Total
currency	Currency code (ex. 'USD')

## 25. Request Customizations

### *Integration Overview*

The CyberSource SFRA cartridge has built-in custom hooks that can be utilized to customize request data being sent to each Service. This can be utilized to send additional custom data that the core cartridge cannot account for. For example, if you want to include Merchant Defined Data in your Credit Card Authorization Requests, you can use these hooks to achieve this.

The hooks are called in the 'scripts/facade/CardFacade.ds' and 'scripts/facade/TaxFacade.js' scripts. After a request for a particular service is built, but before it is sent to CS, a check for any code registering to the hook 'app.cybersource.modifyrequest' is done. If present, the hook will be called for that specific request. The request object is passed into the hook and the return value of the hook is sent to CS as the final request object. Through this process, you can inject your own data into the request object from custom code you write in a separate cartridge.

### *Implementation*

To customize request objects, register the hook 'app.cybersource.modifyrequest' in your cartridges 'hooks.json' file. An example would look like this, replacing the script path with your own script :

```
{
  "name": "app.cybersource.modifyrequest",
  "script": "./cartridge/scripts/hooks/modifyRequestExample"
}
```

You can copy the 'scripts/hooks/modifyRequestExample' script from this cartridge into your own to use as a template for extending and modifying service request objects. Note, every hook must return a valid request object for the given service. It is recommended that you reference the CybserSource documentation for details on the exact nature of any fields you wish to customize or add. The following hooks are available for you to define in this file:

#### **Modify Request hooks**

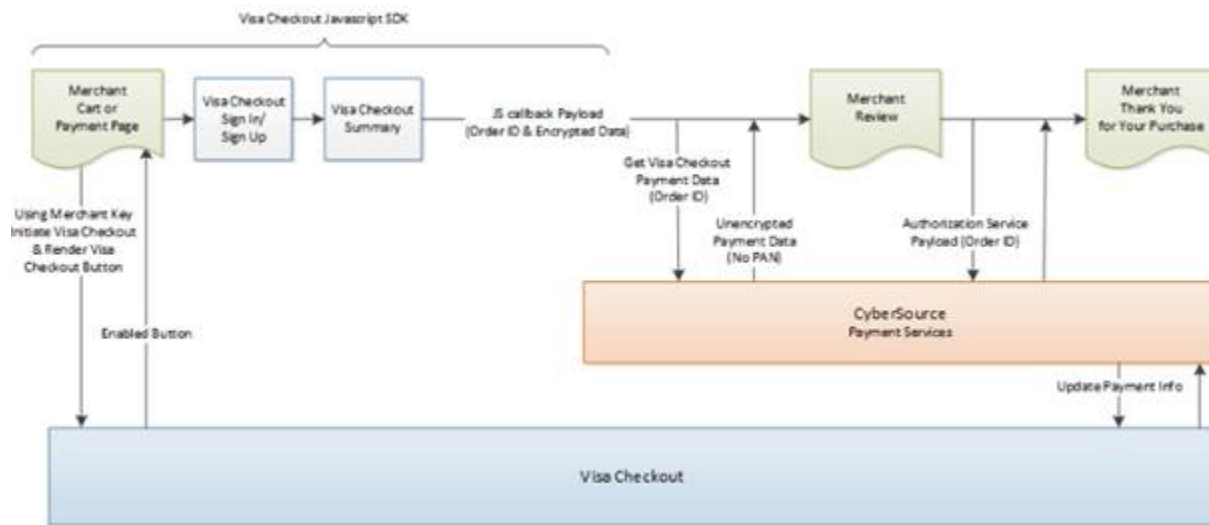
Hook Name	Service Request to modify
CCAuth	Credit Card Authorization
PayerAuthEnroll	Payer Authentication Enrollment
PayerAuthValidation	Payer Authentication Validation
AuthReversal	Credit Card Authorization Reversal
Capture	Credit Card Capture
Credit	Credit Card Credit/Refund
Tax	Tax Calculation

## 26. Visa Checkout

### Integration Overview

Visa Checkout and the CyberSource credit card services work together as an integrated offering. CyberSource provides the following services to assist with your Visa Checkout integration

- Get Visa Checkout data: this service retrieves Visa Checkout data, which enables you to display payment and shipping details to the customer during checkout.
- Authorization: this service enables you to send an authorization request to your processor using the Visa Checkout payment data



1. Your web site integrates directly to Visa Checkout to display the Visa Checkout button on your checkout page.
2. CyberSource provides the get Visa Checkout data service, which retrieves the Visa Checkout payment data, except the PAN. You can use the retrieved data to help the customer confirm the purchase.
3. You submit an authorization request to CyberSource for credit card processing. Instead of including payment information in the authorization request, you include the Visa Checkout order ID.
4. At various points in the transaction cycle, you notify the customer of the transaction status.

### Implementation

To use CyberSource Visa Checkout service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to Visa Checkout services are activated and functioning on your account. To complete the checkout process successfully create Visa Checkout profile on CyberSource business center console under 'Account Management > Digital Payment Solutions > Profiles > Enable Visa Checkout'. Click profile tab and add profile, configure all the mandatory settings, also use API Key from Setting Tab. Create Secret key from

‘Account Management > Client Integration Management > Payment Configuration > Key Management’. Click Generate key and select shared secret. Follow the steps below to configure the service in Business Manager.

1. Determine and set all associated site preference listed below.

## Configuration

### Site Preference Group: CyberSource Secure Acceptance

Preference Name	Usage
cybVisaSdkJsLibrary	Sandbox: <a href="https://sandbox-assets.secure.checkout.visa.com/checkout-widget/resources/js/integration/v1/sdk.js">https://sandbox-assets.secure.checkout.visa.com/checkout-widget/resources/js/integration/v1/sdk.js</a> LIVE: <a href="https://assets.secure.checkout.visa.com/checkout-widget/resources/js/integration/v1/sdk.js">https://assets.secure.checkout.visa.com/checkout-widget/resources/js/integration/v1/sdk.js</a>
cybVisaTellMeMoreLinkActive	Indicate whether Tell Me More Link to be displayed with VISA button true (default) false
cybVisaButtonColor	The color of the Visa Checkout button. standard or neutral.
cybVisaButtonSize	The size of the Visa Checkout button
cybVisaButtonHeight	The height of the Visa Checkout button in pixels.
cybVisaButtonImgUrl	Sandbox: <a href="https://sandbox.secure.checkout.visa.com/wallet-services-web/xo/button.png">https://sandbox.secure.checkout.visa.com/wallet-services-web/xo/button.png</a> LIVE: <a href="https://secure.checkout.visa.com/wallet-services-web/xo/button.png">https://secure.checkout.visa.com/wallet-services-web/xo/button.png</a>
cybVisaCardBrands	Brands associated with card art to be displayed
cybVisaButtonWidth	The width of the Visa Checkout button in pixels.
cybVisaThreeDSSuppressChallenge	Whether a Verified by Visa (VbV) consumer authentication prompt is suppressed for this transaction. If true, VbV authentication is performed only when it is possible to do so without the consumer prompt. true - Do not display a consumer prompt false - Allow a consumer prompt
cybVisaExternalProfileId	Profile created externally by a merchant whom Visa Checkout uses to populate settings
cybVisaSecretKey	The secret key specified VISA Checkout account profile
cybVisaAPIKey	The Visa Checkout account API key specified in cyberSource business center
cybVisaThreeDSActive	Whether Verified by Visa (VbV) is active for this transaction. If Verified by Visa is configured, you can use threeDSActive to deactivate it for the transaction; otherwise, VbV will be active if it has been configured
cybVisaButtonOnCart	CyberSource Visa Button display on minicart and cart

## 27. Bank Transfer

### *Integration Overview*

Bank transfer services enable customers to pay for goods using direct online bank transfers from their bank account to your merchant account. Below are the supported payment methods:

- Bancontact
- EPS
- Giropay
- iDeal
- Sofort

**Bank Transfer Service Support by Country**

Payment Method	Country	Services
Bancontact	Belgium	Sale Check Status Refund
EPS	Austria	Sale Check Status
giropay	Germany	Sale Check Status
iDEAL	Netherlands	Options Sale Check Status Refund
Sofort	Austria Belgium Germany Italy Netherlands Spain	Sale Check Status Refund

Bank transfer supports 4 different services:

- **Option Service:** This service is valid only for iDEAL transactions. The options service (apOptionsService) retrieves a list of bank option IDs and bank names which you can display to the customer on your web site
- **Sale Service:** The sale service (apSaleService) returns the redirect URL for customer's bank. The customer is directed to the URL to confirm their payment details.
- **Check Status Service:** The check status service returns the latest status of a transaction. It is a follow-on request that uses the request ID value returned from the sale service request. The request ID value links the check status request to the payment transaction
- **Refund Service:** The refund service request (apRefundService) is a follow-on request that uses the request ID value returned from the sale request. The request ID value links the refund transaction to the original payment transaction

Bank Transfer functionality is specific to PMs with sale and check status service.

SFRA makes the call to CyberSource Sale service to authorize the purchase amount. A secondary call is made to check a Status service to determine result of the authorization, and the subsequent Order creation or failure.

## Implementation

To use CyberSource Alipay service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to ensure Alipay services are activated and functioning on your account. Bank Transfer supports 5 types of Payment methods – SOFORT, BANCONTACT, IDEAL, GIROPAY & EPS.

Determine and set all associated site preference listed below.

## Configuration

### Site Preferences Attributes

Site Preferences	Description
Merchant Descriptor Postal Code(merchantDescriptorPostalCode)	Merchant Descriptor Postal Code
Merchant Descriptor(merchantDescriptor)	Merchant Descriptor
Merchant Descriptor Contact(merchantDescriptorContact)	Merchant Descriptor Contact
Merchant Descriptor State(merchantDescriptorState)	Merchant Descriptor State
Merchant Descriptor Street(merchantDescriptorStreet)	Merchant Descriptor Street
Merchant Descriptor City(merchantDescriptorCity)	Merchant Descriptor City
Merchant Descriptor Country(merchantDescriptorCountry)	Merchant Descriptor Country

### Payment method attributes

Attribute ID	Description
isBicEnabled	Attribute to check if BIC field is required for EPS and GIROPAY to display on billing page
isSupportedBankListRequired	Attribute to check if bank list is required for IDEAL to display on billing page

BANK\_TRANSFER Details

€

to

¥

to

\$

to

Custom

Payment Type:

EPS (EPS)

Bank Transfer Options

Is Supported Bank List Required:

☐

Is BIC Enabled:

☒

Cybersource Credentials

merchantID:

accenture\_cybersource\_2

merchantKey:

HraLODYZB6jFCxOWnOramcHYn2YTzoYyXSD7/qUB+mpRzZHnIRZzYwM7qtu1MYQEDBBaV5ROGJzsgOdogli7euT+pT1bW2IHeryIPCDHFM8jJlQNPTDFHl.dabQdq57NcFoFp16C5

Apply

Cancel

**Bank Transfer sequence flow:**

1. Select SOFORT or BANCONTACT payment methods from billing page and proceed with the payment
2. For IDEAL, select bank from bank list, for EPS and GIROPAY, enter BIC number and proceed with the payment
3. Creates CyberSource sale service request using bill-to, item data, purchase total data and merchant descriptor data from the current basket
4. Make actual service call to CyberSource sale service
5. If service returns 'ACCEPT' as decision, 100 as reason code and 'pending' as payment status, redirect the user to bank site
6. If service returns 'ACCEPT' as decision, 100 as reason code and 'failed' as payment status, exit immediately, redirect back the user to merchant site along with error message and change the status of order to failed
7. If service returns 'REJECT' as decision, exit immediately, redirect back the user to merchant site along with error message and change the status of order to failed
8. If service returns 'REVIEW' as decision, complete the order transaction but order status would be created itself
9. After successful authorization of amount on bank site, user would be redirected back to merchant site. After redirection, call to CyberSource check status service would be made to complete the transaction
10. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'authorized' or 'settled' as payment status, complete the order and modify order and export status
11. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'pending' as payment status, complete the order without modifying order and export status
12. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'abandoned' or 'failed' as payment status, exit immediately and change the status of order to failed
13. If check status service returns 'REJECT' or 'ERROR' as decision, exit immediately and change the status of order to failed
14. If check status service returns 'REVIEW' as decision, complete the order transaction but order status would be created itself

## 28. Alipay

### *Integration Overview*

The Alipay authorization service allows the storefront application to request an authorization for the order total amount along with the currency. The process begins with a web service call to the CyberSource Alipay Initiate service, which initiates the payment request and potentially authorizes the requested amount. After this request, another call is made to the Check Status service, whose response determines if the order is placed or failed.

Alipay Authorization Sequence Flow:

1. Create CyberSource Alipay Initiate request using purchase total data, product name, and product description (optional) from the current order object.
2. Set Alipay payment type to domestic or international based on site preference.
3. Validate the reason code and Decision of the Initiate request and accordingly set the corresponding variables.
4. If initiation is successful, assign the required values in Demandware Payment Transaction object and create CyberSource Alipay Check Status request using Request ID of Initiate service response.
5. Make service call to Alipay Check Status request to return the payment status of the Initiate request.
6. Validate Reason Code and Payment status of check status service response and set the corresponding variables.
7. If ReasonCode = 100 then check the payment status. If payment status is COMPLETED for service call then complete the checkout flow and place the order with “New” as order status and “Paid” as order payment status.
8. If ReasonCode = 100 and PaymentStatus = PENDING, complete the checkout flow with order status as “Created” and order payment status as “Not Paid”.
9. If ReasonCode = 100 and PaymentStatus = ABANDONED or PaymentStatus = TRADE\_NOT\_EXIST, fail the order and show message on the screen.
10. If Decision = REJECT and ReasonCode = 102 or ReasonCode = 233, fail the order and show message on the screen.
11. If Decision = ERROR and ReasonCode = 150, fail the order and show message on the screen.

Note: As Alipay live environment is not available, so for Alipay Domestic and International scenarios, Site Preference configuration for Reconciliation ID needs to configure to test various scenarios of Alipay Initiate and Check Status service. Also, if shopper does not return from AliPay the SFCC order status will remain in the “Created” until the Batch Job for Check Payment Status service runs.



## Implementation

To use CyberSource Alipay service on SFRA, ensure you have followed all steps in the "Cartridge Installation" guide above. A CyberSource Merchant ID, and CyberSource Merchant Key are required for this service. Enter these values in the corresponding site preferences under the "CyberSource" group and work with CyberSource to ensure Alipay services are activated and functioning on your account.

1. Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip into your sandbox
2. Under 'Merchant Tools > Ordering > Payment Methods' Make sure the 'ALIPAY' payment method is enabled and configured to use the 'CYBERSOURCE\_ALIPAY' payment processor
3. Determine and set all associated site preference listed below.

## Configuration

Site Preference Group: CyberSource Core

Preference Name	Usage
apPaymentType	Alipay Payment Type for Domestic as well as International Payment
apTestReconciliationID	Test Reconciliation ID for Alipay

## 29. Google Pay

### *Integration Overview*

Google pay payment option enables user to pay from google wallet. Google Pay integration with CyberSource provides 3 ways for checkout

1. Cart
2. Mini Cart
3. Billing page

Function flow for Cart and mini cart is similar as compared to Billing page and When user clicks on Google pay button, a google pay window open up for user to enter to login and choose the payment card and billing and shipping address. Whereas in billing page user should have to select only payment card and continue. And then user redirected into Order Review page for completion.

During place order Google Pay payment card data is sent to CyberSource for authorization and authorization status is retrieved from CyberSource.

### *Implementation*

Before proceeding to BM configuration user has to Merchant Account needs to be created with Google please follow below link to create the merchant account with google.

URL : <https://support.google.com/paymentscenter/answer/7161426?hl=en>

- Import 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip

### Business Manager Configuration

1. Go to: "Merchant Tools > Site Preferences > CyberSource Google Pay
2. Check "enableGooglePay"
3. Value for "googlePaygatewayMerchantId" is cybersource Merchant ID
4. Value for "googlePayMerchantID" is the google merchant ID

### *Configurations*

Site Preferences: Cybersource\_GooglePay

Preference Name	Usage
enableGooglePay	Enable or Disable Google Pay Service
googlePayMerchantID	Cybersource Merchant account ID
googlePaygatewayMerchantId	Matching setting on Google Account

### *Integration Overview*

The Klarna authorization service allows storefront application to request for credit authorization for the total order amount. The process begins with initially making the call to CyberSource Init Session service to initialize the Klarna widget, update the klarna session with PII data and Klarna JS API authorization call along with authorization web service call to CyberSource authorization service and receive confirmation about the availability of the funds.

The Demandware KLARNA\_CREDIT–Authorize populates the authorization request with ship-to, bill-to, Klarna Item data, and purchase total data from the basket and invokes the authorization web service call using CyberSource web service API.

Klarna sequence flow:

1. Creates CyberSource Init session request using bill-to data containing only country, state code and postal code and item data and purchase total data from the current basket.
2. Make actual service call to CyberSource Init session service
3. If service returns ACCEPT as decision and 100 as reason code, get the processor token and request id from session service response and set its value into a session variables.
4. If service returns any other decision apart from ACCEPT and 100 as reason code, display an error message on billing page
5. Pass the value of processor token Klarna JS API to load the Klarna widget on summary page
6. Before Klarna JS API authorization call, Create CyberSource Update Session request using ship-to data, bill-to data, item data, and purchase total data from the current basket and request id of init session service response stored in session variable.
7. Make actual service call to CyberSource Update session service.
8. If service returns ACCEPT as decision and 100 as reason code, get the processor token from session service response and set its value into a session variable.
9. If service returns any other decision apart from ACCEPT and 100 as reason code, display an error message on billing page.
10. Create CyberSource authorization request using ship-to, bill-to, item data, and purchase total data from the current basket
11. If Decision Manager is configured in site preference, pass its value to true else false in CyberSource authorization call
12. Click Pay button to first authorize the request through Klarna JS API and then pass the pre-approved token returned by JS API authorization request in CyberSource authorization request
13. If authorization service returns 'ACCEPT' as decision, 100 as reason code and 'authorized' or 'pending' as payment status and If merchant URL redirection is configured in site preference, redirect the user to merchant URL and return back to merchant site to complete the order
14. If authorization service returns 'ACCEPT' as decision and 100 as reason code, 'authorized' as payment status and merchant URL redirection is false, complete the order and modify order and export status

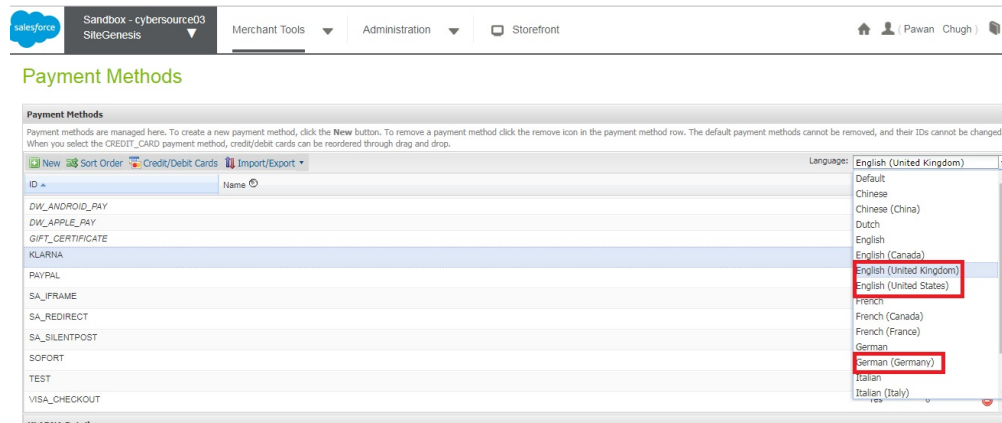
15. If authorization service returns 'ACCEPT' as decision and 100 as reason code, 'pending' as payment status and merchant URL redirection is false, CyberSource check status service would be called to complete the transaction
16. If authorization service returns 'ACCEPT' as decision, 100 as reason code and 'failed' as payment status, exit immediately and change the status of order to failed
17. If authorization service returns 'REJECT' or 'ERROR' as decision, exit immediately and change the status of order to failed
18. If authorization service returns 'REVIEW' as decision, complete the order transaction but order status would be created itself
19. If payment status is 'pending', CyberSource check status service call would be made for both merchant URL redirected orders and non-redirectioned orders
20. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'authorized' or 'settled' as payment status, complete the order and modify order and export status
21. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'pending' as payment status, complete the order without modifying order and export status
22. If check status service returns 'ACCEPT' as decision, 100 as reason code and 'abandoned' or 'failed' as payment status, exit immediately and change the status of order to failed
23. If check status service returns 'REJECT' or 'ERROR' as decision, exit immediately and change the status of order to failed
24. If check status service returns 'REVIEW' as decision, complete the order transaction but order status would be created itself

Validate authorization reason code and set corresponding values, based on Auth response code.

### Merchant Id/Key Specific Changes for Klarna

Different countries and specific currencies could be configured to run Klarna with different Merchant Id/Key specific to different sites. Functional flows would be similar on different sites. Merchant Id/Key could be configured at Merchant Tools -> Ordering -> Payment Methods -> Klarna. In this release, Klarna has been supported for US, UK and Germany with different sites and corresponding Merchant Ids/Key. To update the value of merchant Id/Key specific to the sites, follow below mentioned steps.

On the Payment Methods page, Select the locale (language) you want to set up, then select the Klarna payment method.



Enter the merchantID and merchantKey field in CyberSource Credentials section of payment method. If you leave these empty, the service will fall back to the values you entered in the CS core site preferences.

Next, select the appropriate bill-to language setting under the 'Klarna' custom attribute group.

## Business Manager Configuration

- Ensure you have imported 'CS SFRA PaymentMethods v20\_1.xml' from /metadata/sfra\_meta.zip
- Ensure you have imported 'CS SFRA MetaData v20\_1.xml' from /metadata/sfra\_meta.zip

1. Go to: "Merchant Tools > Site Preferences > CyberSource Klarna
2. Check " Klarna Decision Manager Required".
3. The Value for " Klarna JS API Path" is Klarna JS API Library Path.

## Configurations

### Site Preferences: Cybersource\_Klarna

Preference Name	Usage
Klarna Decision Manager Required	Enable or Disable Decision Manager
Klarna JS API Path	Klarna JS API Library Path

## 31. Misc

Failover/Recovery Process: Visa has dedicated data centers in Virginia and Colorado. There are no single points of failure. Visa Data Centers implement redundant, dual-powered equipment, multiple data and power feeds, and fault tolerance at all levels with 99.995% uptime. In case of any failover, please open support case @ <https://support.cybersource.com>

Supported Locales: Out of box cartridge supports most of the locales like English (United States), English (United Kingdom), French (FRANCE), English (Austria), German (GERMANY), Dutch (NETHERLANDS) etc. For complete list of supported locales for Cybersource Secure Acceptance Hosted checkout, refer [https://apps.cybersource.com/library/documentation/dev\\_guides/Secure Acceptance Hosted Checkout/Secure Acceptance Hosted Checkout.pdf](https://apps.cybersource.com/library/documentation/dev_guides/Secure%20Acceptance%20Hosted%20Checkout/Secure%20Acceptance%20Hosted%20Checkout.pdf)

# Release History

Version	Date	Changes
18.1.0	10-25-2018	- Initial SFRA release.
18.1.1	11-6-2018	<ul style="list-style-type: none"> <li>- Adds hooks to customize request objects.</li> <li>- Separates subscription creation option to use a new site preference.</li> <li>- Adds Facade for 'Credit Card Credit' Service.</li> <li>- Adds Facade for 'Credit Card Capture Service.</li> <li>- Adds Facade for 'Credit Card Auth Reversal' Service.</li> </ul>
18.1.2	1-9-2019	<ul style="list-style-type: none"> <li>- Update documentation with SFRA compatibility note.</li> <li>- Remove public facing endpoints that can finalize an Order.</li> <li>- Fix to SA SilentPost not processing Orders with Fraud status of 'Review'</li> <li>- Ensure all Orders with Fraud decisions of 'Review' are placed in a 'Not Confirmed' state.</li> <li>- Utilize CS endpoint preference to set Test or Production url endpoints.</li> <li>- Always Send Date to Flex Token API in US English format.</li> </ul>
19.1.0	2-8-2019	<ul style="list-style-type: none"> <li>- Adds PayPal and PayPal Express Integrations</li> <li>- Security patch to Test suite in Controllers Cartridge.</li> <li>- Update to SA Flex date generation to require an en_US local.</li> <li>- Fix to SA Flex payment response data being saved properly.</li> </ul>
19.1.1	6-6-2019	<ul style="list-style-type: none"> <li>- PayPal will now utilize the 'CyberSource Endpoint' site preference.</li> <li>- Update to accommodate hitting the back button during SA redirect checkout.</li> <li>- Added details to documentation.</li> </ul>
19.3.0	7-26-2019	- Update 3DS to version 2.0 utilizing Cardinal Cruise.
19.3.1	09-10-2019	- 3DS Documentation update
19.3.2	09-13-2019	- Bug fix on basic credit transactions to work as when Cardinal Cruise/Payer Auth is not configured
19.3.4	11-14-2019	- Supports Klarna payment and replace Conversion Detail Report to REST API
20.1.0	1-31-2020	<ul style="list-style-type: none"> <li>- Updated cartridge to make it compatible with 4.3.0 version of SFRA.</li> <li>- Bug fix on silent Post Payment method.</li> <li>- Change reference code from "test" to unique ID in subscription creation and deletion call.</li> <li>- Bug fix on JCB and Dinner club card processing.</li> <li>- Minor fixes on Tax calculation that uses CyberSource tax services.</li> <li>- Improved error handling on SA Flex and SA iframe.</li> <li>- Bug fixes on PayPal button logo in Germany locale.</li> <li>- Restructured cartridge folder to adhere to Salesforce's new standards.</li> </ul>