Salesforce Commerce Cloud – B2C Commerce Extension from Fiserv Payments

V 1.0.0

Introduction

Carat is the global commerce platform from Fiserv that orchestrates payments and experiences for the world's largest businesses.

Commerce Hub is the modernized payment technology stack from Fiserv that facilitates the enablement of payments and value-added services for omni commerce. A connection to Commerce Hub will afford any merchant a single solution to meet their growing commerce needs.

Salesforce Extension Overview

For the first time, a Salesforce Commerce Cloud- B2C Commerce extension has been created for Carat from Fiserv Payments. This is a new initiative from Fiserv where we want to make Carat and Commerce Hub easily accessible to merchants across all categories and geographies.

Our intentions with this product are twofold:

1) We are building this extension to be fit for purpose.

   Fiserv will serve a merchant's needs as their business grows. At launch, you may just need domestic card processing. Soon after, you desire to offer a better experience for your customers so tokenization and card-on-file will become a requirement. Your business expands into new regions. Fiserv can enable additional currencies and additional payment methods applicable to those markets.

   Wherever you are in your growth journey, Fiserv will fit your purpose wherever you are.

2) We want to make payments simple and fast.

   Fiserv has a strong legacy in payments and understands not only the complexity of payments, but also the regulatory landscape. Given our position in the market, we are on a mission to simplify payments for you so that you can accomplish your commerce goals at speed; hence, our motivation to develop our Salesforce B2C Commerce extension.

   While initial market entry of this will offer fundamental features, our commitment is to keep parity with the rich solutions that Carat offers. You can expect additional iterations soon.

Integration Guide
June 2025
V 1.0.0

Establish Account & Pricing

Please contact your sales representative or relationship manager.

Business Use Cases

Current Business Use Cases

The supported business cases focus on enabling credit and debit card processing for card-not-present scenarios. At the point of purchase (e.g. online checkout) by the consumer, the extension facilitates the following.

Credit & Debit Cards

- Supports a consumer to present a credit or debit card to purchase goods and services as a "sale".
- Supports a consumer to present a credit or debit card to purchase goods and services as an "authorization only" flow.
- Supports the opt-in or opt-out tokenization of cardholder data as a stored payment method for future use.
  - Supports the addition or deletion of store payment methods. This is a self-management tool given to the consumer within their account profile of the merchant storefront.
- Supports the input and capture of cardholder data through a PCI compliant and customizable implementation where sensitive data is hosted by Fiserv. This is rendered during consumer checkout via an iFrame provided by Commerce Hub's Checkout solution.
- Supports split tender between credit / debit cards and gift cards. Only one credit/debit card can be applied towards a given order. However, up to a max of 10 gift cards can be applied to the same order.

Fiserv Gift Cards

- Supports a consumer to perform a balance inquiry during checkout while applying a gift card as a form of payment. The balance will be applied to the shopping cart total.
- Supports a consumer to perform a balance inquiry only flow independent from checkout for those scenarios where a consumer is allowed to check their gift card balance.
- Supports a consumer to perform a gift card redemption
  - For the full amount of the purchase
  - For a partial amount of the purchase in combination with a credit/debit card
- Supports a consumer to present a gift card to purchase goods and services as a "sale".
- Supports a consumer to present a gift card to purchase goods and services as an "authorization only" flow
- Ability to input and capture gift card data through the same PCI compliant and customizable implementation as credit / debit cards fields are hosted by Fiserv. This is rendered during consumer checkout via an iFrame provided by Commerce Hub's Checkout Hosted Pages. While

gift cards do not fall under PCI scope, this provides for a consistent integration and checkout experience for consumers.

- Enables multi-tender support – credit / debit card, gift card
    - Only one credit/debit card can be applied towards a given order. However, up to a max of 10 gift cards can be applied to the same order.
    - In a multi-tender scenario, gift cards are always authorized before authorizing a card payment method. If the subsequent card auth is declined, the extension will automatically void the gift card authorization. If a sale is configured, a merchant must then refund the amount based on their Standard Operating Procedures.

Features and Functionality

Countries & Currencies

- United States (domiciled merchants)
- USD

Enablement of additional countries will be enabled over time. Please note that with this current extension, merchants can accept purchases from other countries, but the prices and amounts will be presented in USD. Any transactions purchased from abroad will incur foreign exchange fees which will be borne by the cardholder.

Payment Methods

- Visa
- Mastercard
- American Express
- Discover
- Diners Club
- JCB
- Fiserv Gift Cards

Payment Functionality

- Authorization only
  This is a function to reserve a defined amount on the consumer's payment method with the intention of sending a subsequent capture request to fully charge and settle funds.

- Sale (authorization + capture)
  This is a function that combines authorization and capture together in a single operation. This will remove the need to send a subsequent capture call and normally used for products that are immediately delivered to the consumer – access to content, downloadable software, etc.

- Tokenization
  This enables a form of payment to be tokenized and stored in the consumer's profile for card-on-file use cases. When enabled, the consumer can opt to store their card for future use.
  NOTE: This is only available for credit and debit cards – not gift cards.

- Fiserv Gift Cards
  - Balance Inquiry
  - Redemption
    - The redemption function has been built to follow the traditional card flows for auth and capture to be consistent with customary process flows. A sales flow can be configured as well where an authorization and capture can be executed in a single request.

- Split tender / multi tender
  - For a given order purchase, a consumer is able to split their payment across two payment methods – a credit/debit card and a gift card
  - The use of 1 credit/debit cards and/or up to 10 gift cards can be used per order.

Note

- Salesforce B2C Commerce does not come with an order management system natively built into it. In order to carry out secondary transactions such as Voids, Refunds, Captures etc. or order fulfillment functionality, these will need to be carried out by a 3rd party Order Management System. Onus is on the Merchant to facilitate the Order fulfillment functionality for their business.

Security & PCI Compliance

Fiserv understands the importance of mitigating PCI risk for merchants and has taken liberty to leverage Commerce Hub's Hosted Fields. The plugin enables the use of an SAQ A iFrame integration to facilitate the collection of sensitive card information as well as offering field customization to better match merchant website branding.

Salesforce Commerce Cloud - B2C Commerce Platform Compatibility

Edition:

Storefront Architecture

Salesforce B2C Commerce Version:

SFRA 7.0, Compatibility mode 22.7 (SFRA_AP_07_10_2024)

Fiserv Checkout SDK

Version 3.1.21

Browers Tested (Storefront and Admin):

Chrome
Firefox
Microsoft Edge
Safari

Quick Start

The first order of business is to install, configure, and run your first transaction in our test (e.g. CERT) environment. The extension is built ready-to-go. The installation has been constructed to have pre-configured settings to run a transaction within minutes upon configuring your test account and credential information.

Test Account and Credentials

To run your first transaction, you need to establish a test account and credentials through Dev Studio. If this has not been established previously, follow these quick instructions.

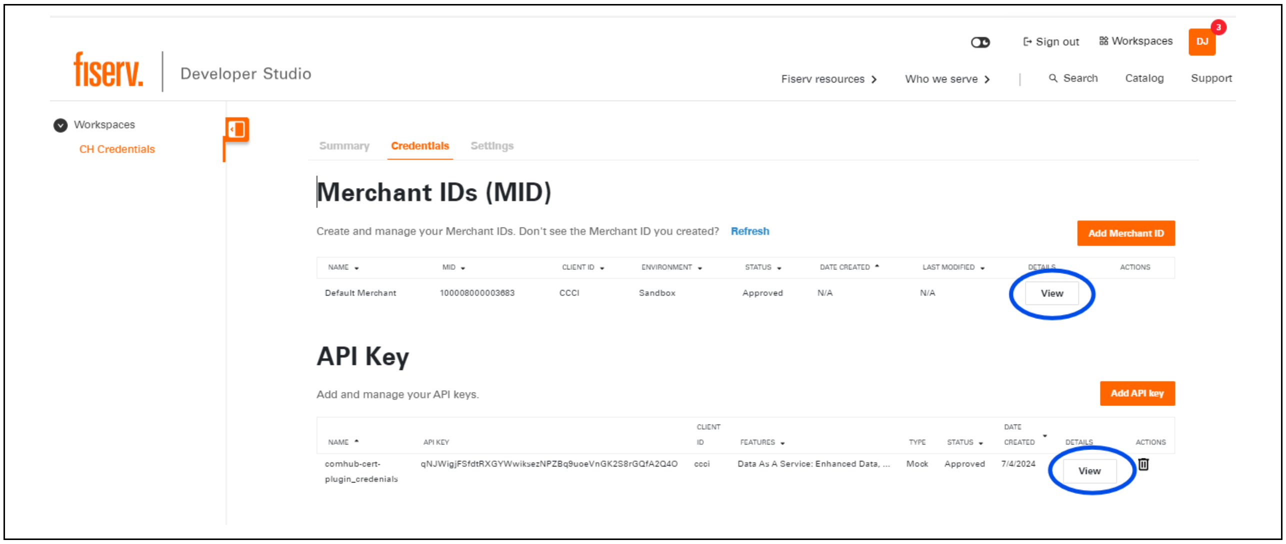
1) Navigate to the Fiserv Developer Studio (Dev Studio-- https://developer.fiserv.com/)
2) Create an account / login
3) Create a workspace
    a. Create a Merchant ID and Termina ID (e.g. MID and TID)
    b. Create API Key and API Secret

All information about how to execute the above can be found at:
https://developer.fiserv.com/support/docs/?path=docs/guides/workspaces.md

As a result of these steps, you will have access to the following:

- Merchant ID
- Terminal ID
- API Key
- API Secret

You will need these four values to run your first transaction. Please save them in a convenient location. They can be found in the Credentials tab of your Dev Studio Workspace.



Merchant IDs (MID)

Create and manage your Merchant IDs. Don't see the Merchant ID you created? Refresh

NAME	MID	CLIENT ID	ENVIRONMENT	STATUS	DATE CREATED	LAST MODIFIED	DETAILS	ACTIONS
Default Merchant	100008000003683	CCCI	Sandbox	Approved	N/A	N/A	View	

API Key

Add and manage your API keys.

NAME	API KEY	CLIENT ID	FEATURES	TYPE	STATUS	DATE CREATED	DETAILS	ACTIONS
comhub-cert-plugin_credenials	qNJWigjFSfdtRXGYWwiksezNPZBq9uoeVnGK2S8rGQfA2Q4O	ccci	Data As A Service: Enhanced Data, ...	Mock	Approved	7/4/2024	View	

Integration

Integration Playbook

As a guide to an efficient integration, we suggest the following actions and their order of operations.

1) Begin securing a LIVE PRODUCTION account – e.g. MID and TID. Please contact sales or if you are an existing client, your relationship manager
    a. This step will start the any commercial and business requirements – including pricing, contracting, and onboarding
    b. CERT accounts are automatically provisioned through Dev Studio
2) Secure CERT accounts and credentials through Dev Studio
3) Install Salesforce Commerce Cloud- B2C Commerce extension from Fiserv Payments
4) Configure Salesforce extension
5) Run first transaction as a proof of concept that payment extension is working
6) Configure extension for your business requirements
7) Implement and Test
    a. Consumer checkout. Integrate extension into your storefront process flows.
    b. Merchant Tools. Integration into Salesforce Merchant Tools Dashboard.
    c. Consumer payment method management. Optional – if enabled.
8) Field customizations
9) UAT
10) End to End
11) Go Live

Step 1: Download & Installation

1) Navigate to Salesforce Business Manager and click Administration-> Site Development -> Import and Export
2) Click upload
3) Upload file from repo: REPO_BASE-> site_import-> metadata-> system-objecttype-extensions.xml
4) Select import under import metadata
5) Select system-objecttype-extensions.xml then import
6) Hit refresh
7) Click next in bottom right (make sure no errors)
8) Return to Administration-> Site Import and Export
9) Zip file from repo: REPO_BASE-> site_import
10) Upload file from repo: REPO_BASE-> site_import.zip
11) Select site_import.zip and import
12) Confirm you want to import (make sure no errors)
13) Go to VS Code and open up the repository
14) Change your dw.json instance link to point at the current sandbox
15) Open prophet in VS Code
16) Click the 3 dots next to cartridges then select 'Prophet: Enable Upload'
17) Once upload is complete, select 'Prophet: Disable Upload'
18) Go to Administration-> Manage sites
19) Click the site you want to deploy to
20) Click settings
21) Add "fiserv_commercehub:" to the front of the cartridge path
22) Click apply
23) Go back to Administration-> Manage sites
24) Click the link under business manager
25) Add "fiserv_commercehub:app_storefront_base:" to the front of the cartridge path
26) Click apply
27) Restart the sandbox
28) Re-open the business manager
29) Go to Administration-> Roles and Permissions
30) Select Administrator
31) Select Business Manager Modules
32) Select all websites (not Organization)
33) Select all sections under "Fiserv Payments"
34) Scroll to the bottom and select Update
35) Go to Operations-> Custom Log Settings
36) Select box next to DEBUG and INFO
37) Select DEBUG in dropdown next to root category
38) Click save in top right corner

Step 2: Fiserv Payments Configuration Settings

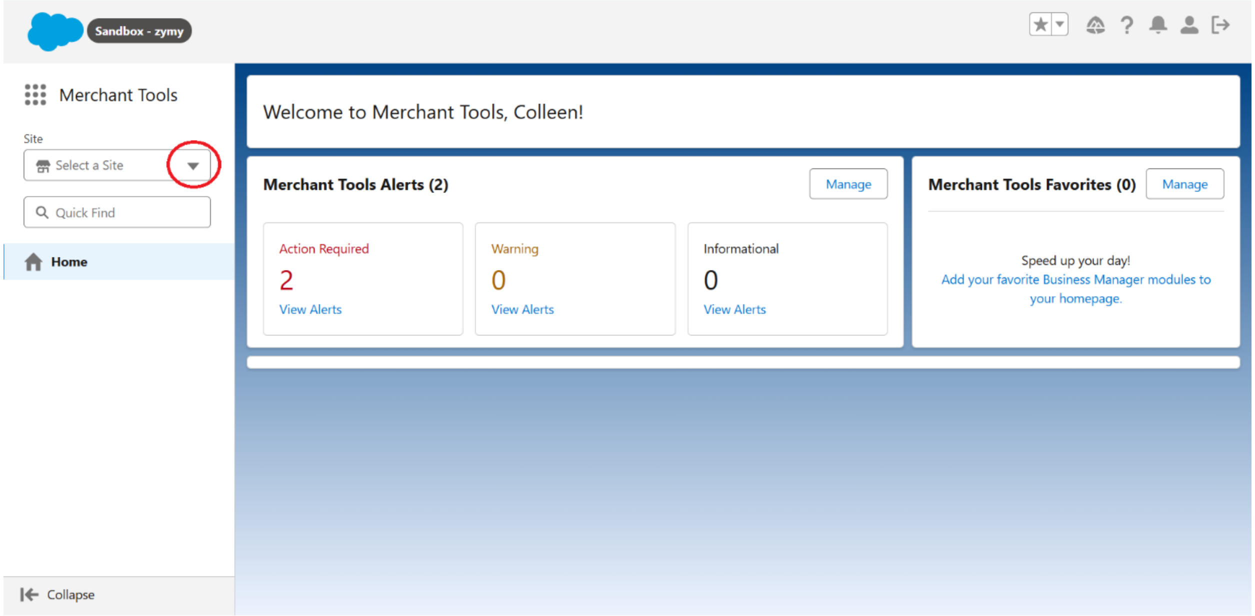
!!! REMINDER: Configuration and first transaction cannot be completed until a test account and credentials have been created. See Quick Start section (above).

Upon downloading and installing the Salesforce extension from Fiserv Payments, the next step is to configure the plugin so that we can run a first test transaction. This will ensure that all initial environmental and configuration settings are in place.

Below are steps to navigate to the Fiserv Payments Configuration Settings. Within the Salesforce Business Manager Dashboard, navigate to:

Site > ‘Ref Arch’

Navigate to Ref Arch Site

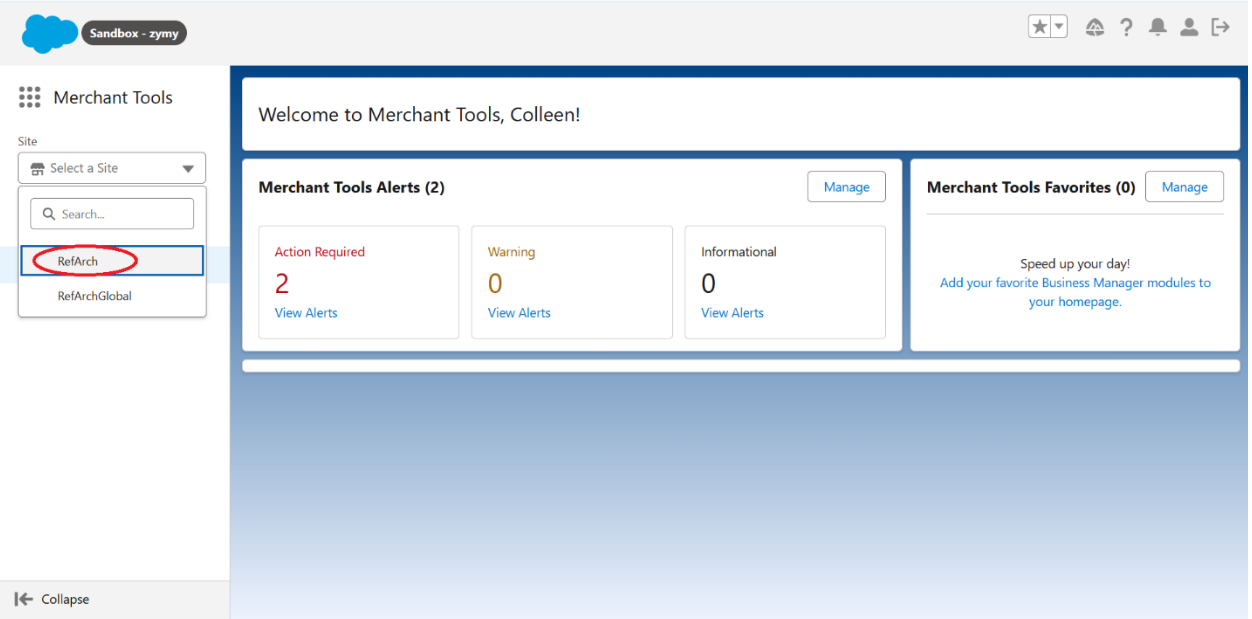


Welcome to Merchant Tools, Colleen!

Merchant Tools Alerts (2)

Merchant Tools Favorites (0)

Speed up your day!
Add your favorite Business Manager modules to your homepage.



Welcome to Merchant Tools, Colleen!

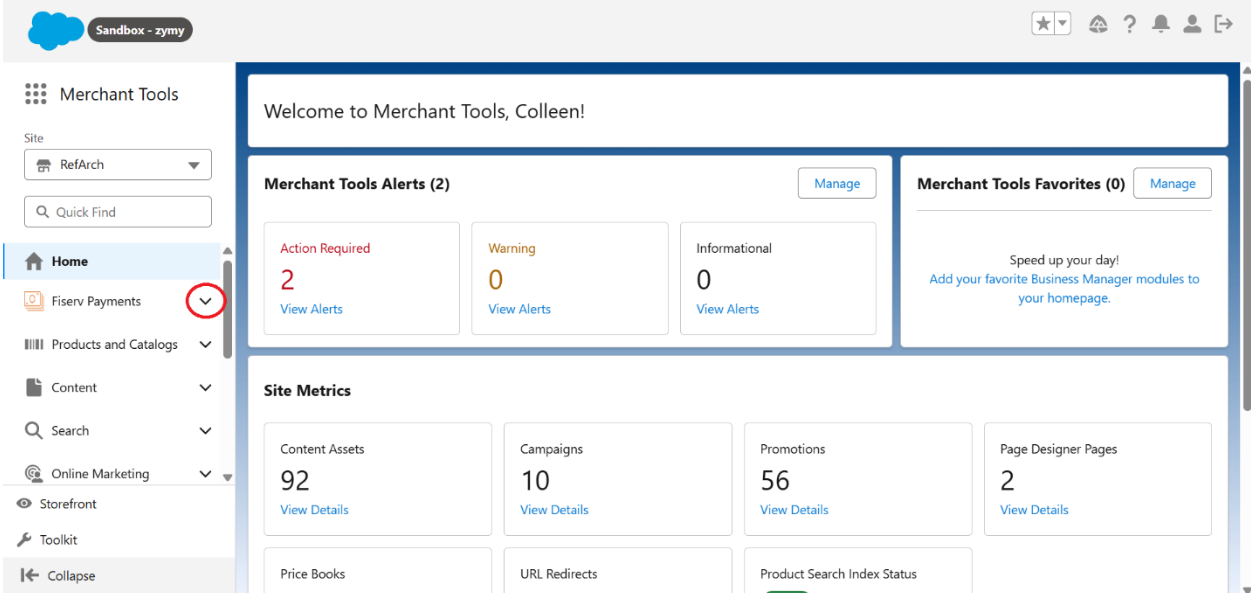
Merchant Tools Alerts (2)

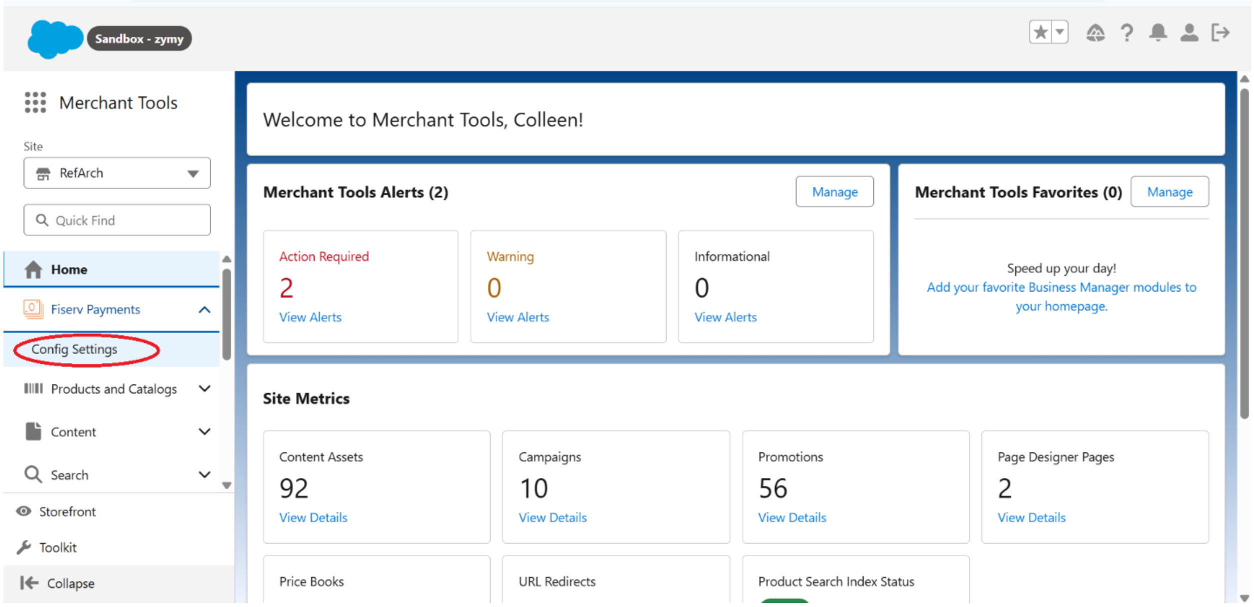
Merchant Tools Favorites (0)

Speed up your day!
Add your favorite Business Manager modules to your homepage.

Navigate to Config Settings

Fiserv Payments > Config Settings





All configurations will be done on this page. The configuration section has been categorized by the following:

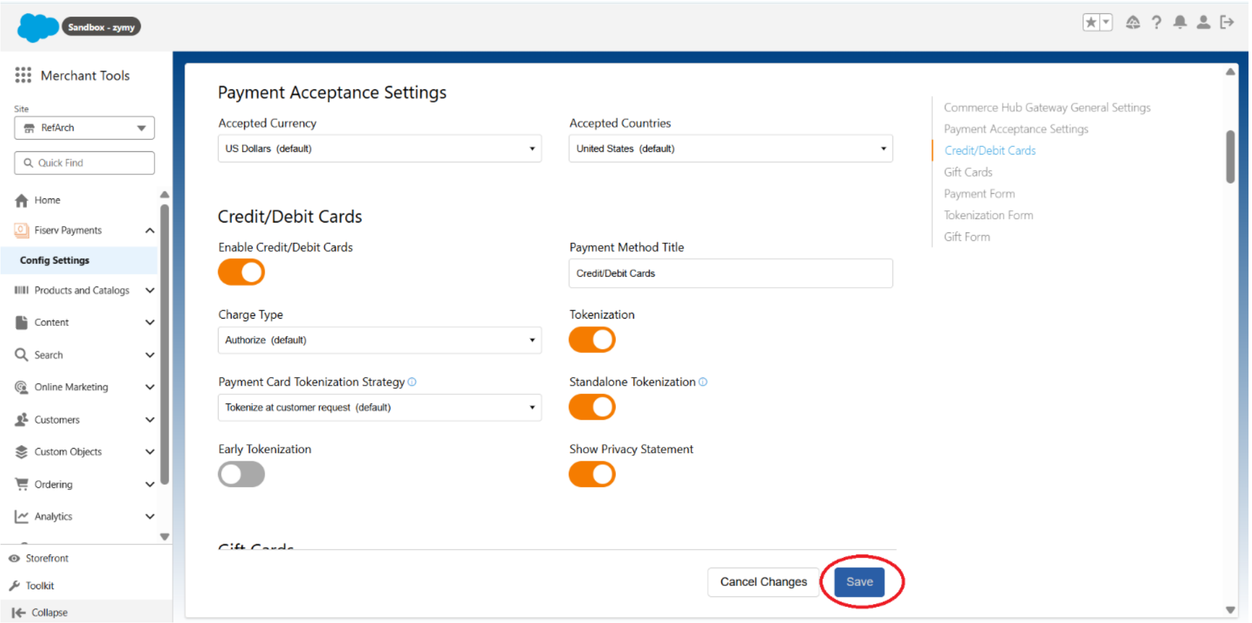
- Commerce Hub General Settings
- Payment Acceptance Settings

- Credit/Debit Cards
- Fiserv Gift Cards
- Payment form Customization
- Tokenization form Customization
- Gift form Customization

Details of these categories are explained in Step 3.

Saving Configurations

Each time a configuration change is made, one must remember to save changes. To do this, you must choose “Save Config” located at the bottom of the page. This must be completed for changes to take effect.



Payment Acceptance Settings

Accepted Currency: US Dollars (default)

Accepted Countries: United States (default)

Credit/Debit Cards

Enable Credit/Debit Cards: on

Payment Method Title: Credit/Debit Cards

Charge Type: Authorize (default)

Tokenization: on

Payment Card Tokenization Strategy: Tokenize at customer request (default)

Standalone Tokenization: on

Early Tokenization: off

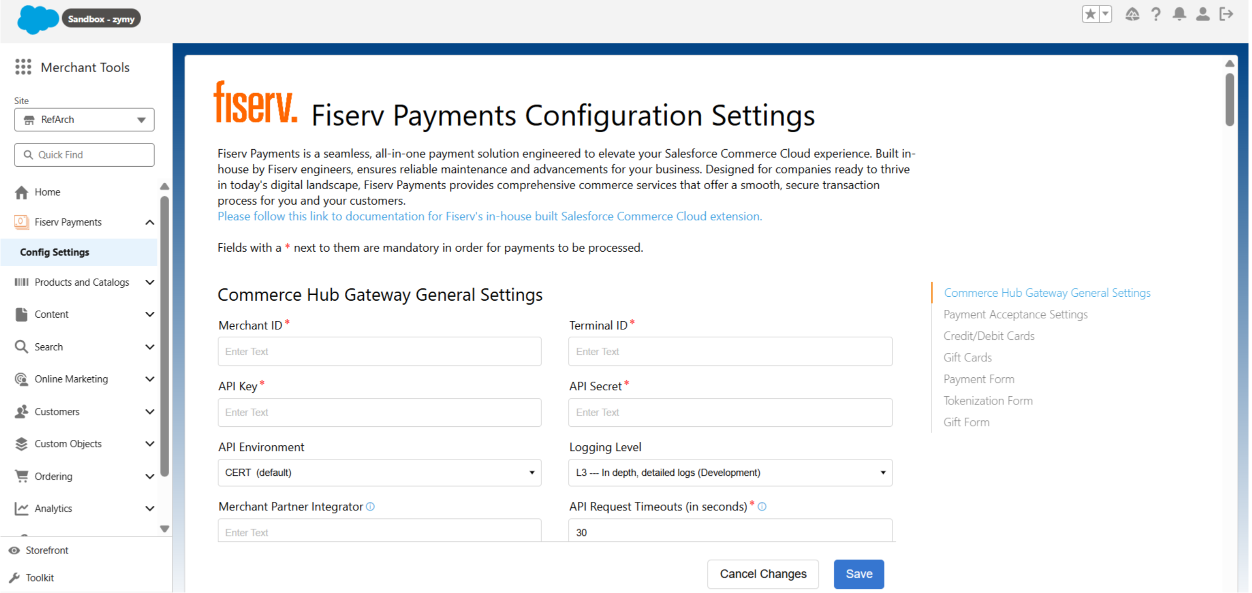
Show Privacy Statement: on

Step 3: Configuration Instructions for Fiserv Payments Configuration Settings

This section explains the various configuration areas, functions, and purpose.

Commerce Hub Gateway General Settings

This area determines the setup required to run transactions with the Adobe extension.



Fiserv Payments Configuration Settings

Fiserv Payments is a seamless, all-in-one payment solution engineered to elevate your Salesforce Commerce Cloud experience. Built in-house by Fiserv engineers, ensures reliable maintenance and advancements for your business. Designed for companies ready to thrive in today's digital landscape, Fiserv Payments provides comprehensive commerce services that offer a smooth, secure transaction process for you and your customers.

Please follow this link to documentation for Fiserv's in-house built Salesforce Commerce Cloud extension.

Fields with a * next to them are mandatory in order for payments to be processed.

Commerce Hub Gateway General Settings

Merchant ID * Enter Text

Terminal ID * Enter Text

API Key * Enter Text

API Secret * Enter Text

API Environment: CERT (default)

Logging Level: L3 — In depth, detailed logs (Development)

Merchant Partner Integrator: Enter Text

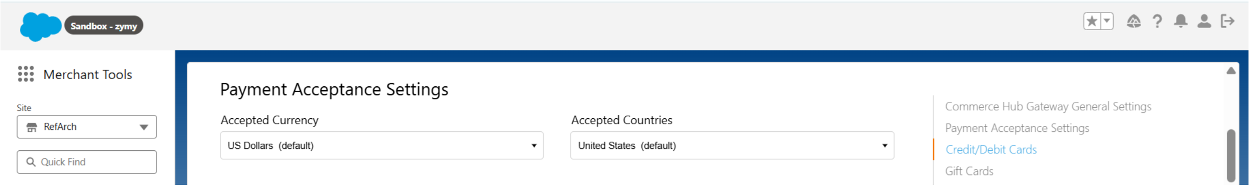
API Request Timeouts (in seconds) *: 30

Configuration Field	Value	Description
Merchant ID	String	Provided through Dev Studio.
Terminal ID	String	Provided through Dev Studio.
API Key	String	Provided through Dev Studio.
API Secret	String	Provided through Dev Studio. **NOTE. Please be mindful that there are different credentials and identifiers associated with the different environments. Please ensure you have the correct ones for both CERT and PROD, respectively.
Merchant Partner Integrator	String	Please enter the entity that integrated the Adobe extension on behalf of the merchant – could be the merchant themselves or a 3rd party systems integrator.
API Environment	Values: • CERT	Fiserv supports two environments – CERT and PROD.

	- PROD	**CERT** is the Carat development and testing environment for merchants and integrators. This is the environment where merchants and integrators will set up, configure, and perform UAT (user acceptance testing) for their end-to-end validation of how their web storefront will interact with the plugin. It is the merchant's responsibility to ensure their workflows and business rules are met. **PROD** is the Carat production environment. Once validation is complete, this is how a merchant implementation of the plugin can be promoted to a production environment for LIVE transactions.
Logging Level	Values: - L1 – Basic information - L2 – Error reporting - L3 – In depth, detailed logs (Development)	See the Transaction Logging section to learn more.
API Request Timeout	String	The amount of seconds the merchant wishes for the API to wait for a response back before issuing a timeout error, set to 30 seconds by default

Payment Acceptance Settings

This area determines the general payment acceptance in the extension.



Payment Acceptance Settings

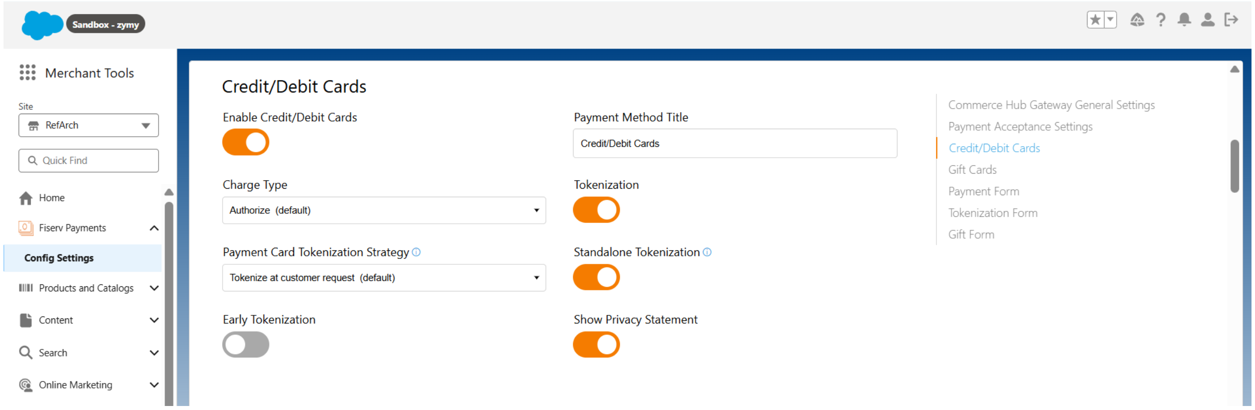
Accepted Currency: US Dollars (default)

Accepted Countries: United States (default)

Configuration Field	Value	Description
Accepted Currency	US Dollar	US Dollar (USD) Only **NOTE. In this version of the plugin, only US Dollar (USD) is enabled. More currencies will be available over time.
Accepted Countries	United States	United States Only **NOTE. In this version of the plugin, only accepted country is the United States. More countries will be available over time.

Credit/Debit Cards

This section defines payment card specific settings.



Credit/Debit Cards

Enable Credit/Debit Cards: on

Payment Method Title: Credit/Debit Cards

Charge Type: Authorize (default)

Tokenization: on

Payment Card Tokenization Strategy: Tokenize at customer request (default)

Standalone Tokenization: on

Early Tokenization: off

Show Privacy Statement: on

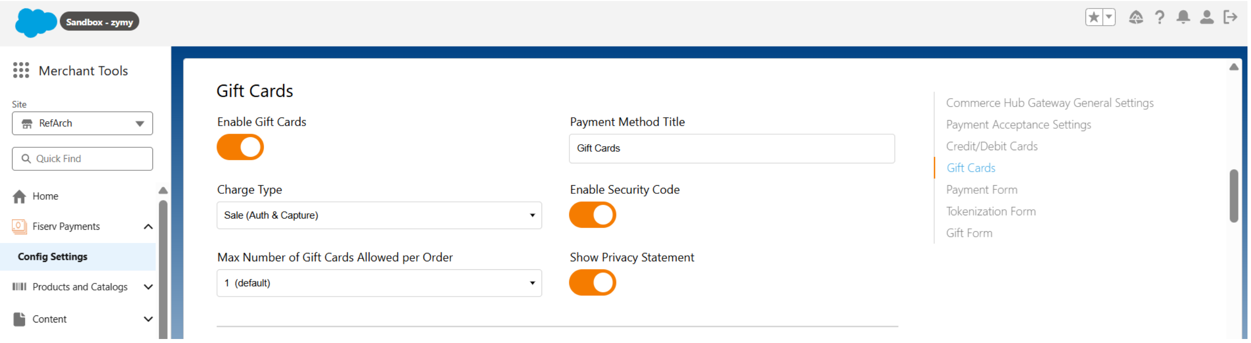
Configuration Field	Value	Description
Enable Credit and Debit Cards	Boolean Toggle: Yes / No	This configuration will enable the acceptance for credit and debit card features, and can be used independently of any other payment method in the SFCC extension. Toggle On, means that acceptance of credit and debit cards are enabled. Toggle Off, means that the acceptance of credit and debit cards are disabled.
Charge Type	Values: • Sale (Authorization + Capture) • Authorize	**Sale (Authorize + Capture)** will fully charge the consumer at the time of checkout. **Authorize** requires a subsequent capture request. This charge type will place a hold / reserve on funds.
Payment Method Title	String	This is the label shown to the consumer for the credit / debit card section of available payment methods.
Tokenization	Boolean Toggle: Yes / No	This configuration will allow consumers to opt-in or opt-out to store (e,g, tokenize) their credit/debit cards within their merchant login profile. For future purchases when logged in, consumers can select a stored card on file rather than entering a new card. Toggle On, means that a check box will be shown at checkout so that a consumer can opt to save their card details; **DEFAULTED to Yes** Toggle Off, means that a check box will NOT be shown at checkout. A consumer card does not have the option of being saved.
Payment Card Tokenization Strategy	Values: • Tokenize all transactions • Tokenize at customer request	This configuration is only enabled when **"Tokenization"** is set to **Yes**. This configuration tokenizes a payment card 100% of the time or only when a consumer requests it.

		Tokenize all transactions enables the Salesforce extension to tokenize all credit / debit card transactions. This is typically used when a merchant required a tokenized form of payment for downstream operational and customer service issues. **Tokenize at customer request only** tokenizes credit / debit cards when the consumer opts-in to save their form of payment. **NOTE. Consumers will only see their stored payment methods in their profile when the opt-in to save their form of payment at the time of purchase.
Standalone Tokenization	Boolean Toggle: Yes / No	This configuration is only enabled when **“Tokenization”** is set to **Yes**. This configuration is tied to the above value. If "Tokenization" = No, then this configuration field is not an option and will not even shown. If available, Toggle On, means that a consumer can log into their merchant store account, navigate to their stored payment method page, and add or delete a payment method (inside of their consumer profile). Toggle Off, means that a consumer CANNOT log into their merchant store account and save or delete a stored payment method.
Early Tokenization	Boolean Toggle: Yes / No	This configuration is tied to the above value. If "Tokenization" = No, then this configuration field is not an option and will not even shown. If available, Toggle On, means that tokenization will occur when the consumers enters their payment credentials in the basket, but before the final order call is placed

		Toggle Off, means that tokenization will occur at usual stage when final order call is placed
Show Privacy Statement	Boolean Toggle: Yes / No	This configuration controls whether the Fiserv Privacy statement is shown under this payment method on the checkout page. Toggle On, means that the Fiserv privacy statement will be shown on the page. Toggle Off, means that the Fiserv privacy statement will not be shown on the page.

Fiserv Gift Cards

This section defines Fiserv Gift Cards specific settings.



Gift Cards

Enable Gift Cards: ●

Payment Method Title: Gift Cards

Charge Type: Sale (Auth & Capture)

Enable Security Code: ●

Max Number of Gift Cards Allowed per Order: 1 (default)

Show Privacy Statement: ●

Configuration Field	Value	Description
Enable Gift Cards	Boolean Toggle: Yes / No	This configuration enables or disables gift card acceptance. Toggle On, enables gift card acceptance. Toggle Off, disables gift card acceptance. **NOTE. This feature requires an additional agreement / addendum for enablement. Please refer to your sales associate or relationship manager.
Payment Method Title	String	This is the label shown to the consumer for the gift card section of available payment methods.

Charge Type	Values: • Sale (Authorization + Capture) • Authorize	**Sale (Authorize + Capture)** will fully charge the consumer at the time of checkout. **Authorize** requires a subsequent capture request. This charge type will place a hold / reserve on funds.
Max Number of Gift Cards Allowed per Order	String	Currently, a max of 10 gift cards are allowed per order. It will be defaulted to “1” but merchant can configure to a max of 10 gift cards per order
Enable Security Code	Boolean Toggle: Yes / No	This configuration controls whether the consumer will need to enter the security code of the gift card on the checkout page. Toggle On, means the consumer must enter the security code of the gift card in payment details Toggle Off, means the consumer does not have to enter the security code of the gift card in payment details
Show Privacy Statement	Boolean Toggle: Yes / No	This configuration controls whether the Fiserv Privacy statement is shown under this payment method on the checkout page. Toggle On, means that the Fiserv privacy statement will be shown on the page. Toggle Off, means that the Fiserv privacy statement will not be shown on the page.

Step 4: Checkout Hosted Field Customizations

To ensure that we protect consumer payment information and help merchants manage their PCI and PII obligations, we have implemented a means to place all payment input data fields into Hosted Fields.

While we have helped manage the capture and input of sensitive payment data at checkout, we do recognize the importance of a consumer's experience with a merchant's brand and therefore have provided individual field customization options.

Customization Architecture

In the Salesforce B2C Commerce extension, there are a handful of touch points where consumer payment data can be entered – namely payment cards and gift cards.

- At Consumer Checkout
- Within the consumer's account profile

We have provided the following customization areas for management control.

1. Payment Form
2. Tokenization Form
3. Gift Form

Customizations leverage Checkout Hosted Fields. Please follow this link to view documentation.

https://developer.fiserv.com/product/CommerceHub/docs/?path=docs/Online-Mobile-Digital/Checkout/Hosted-Fields/Hosted-Fields.md&branch=main

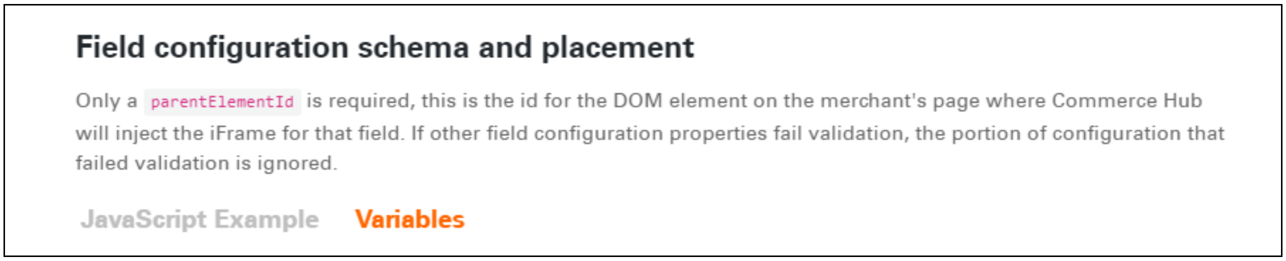
Field Customization

For the Salesforce B2C Commerce extension, we have taken away the complexity and nuances of field customizations by translating this into an easy-to-use interface.

Integration has been provided by the following supported fields:

cardNumber
nameOnCard
securityCode
expirationMonth
expirationYear

With the following field configurations which accommodates both payment cards and Fiserv gift cards (choose Variables tab):



Field configuration schema and placement

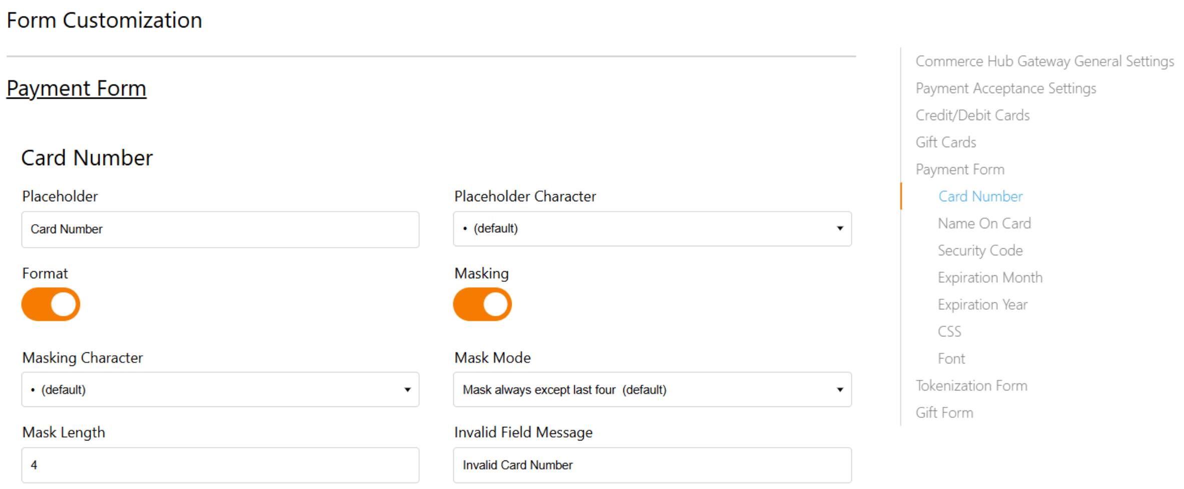
Only a `parentElementId` is required, this is the id for the DOM element on the merchant's page where Commerce Hub will inject the iFrame for that field. If other field configuration properties fail validation, the portion of configuration that failed validation is ignored.

placeholder
dynamicPlaceholderCharacter
enableFormatting
masking.charcacter
masking.mode
masking.shrunkLength

On the same page, see masking modes that are available for those customarily sensitive data fields.

Each supported field will differ. For example, treatment of a credit card field is different than that of an expiration year field. Each field has been provided with options fit for purpose.

See below screenshots of each available field and options.



Form Customization

Payment Form

Card Number

Placeholder: Card Number

Placeholder Character: • (default)

Format: ●

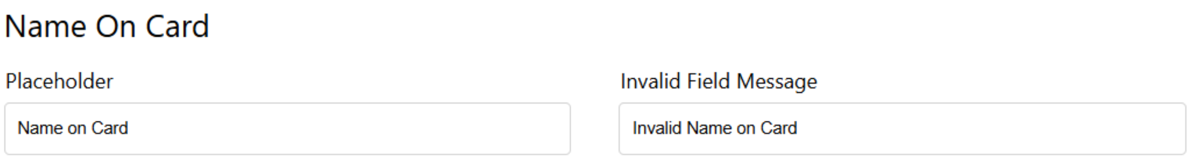
Masking: ●

Masking Character: • (default)

Mask Mode: Mask always except last four (default)

Mask Length: 4

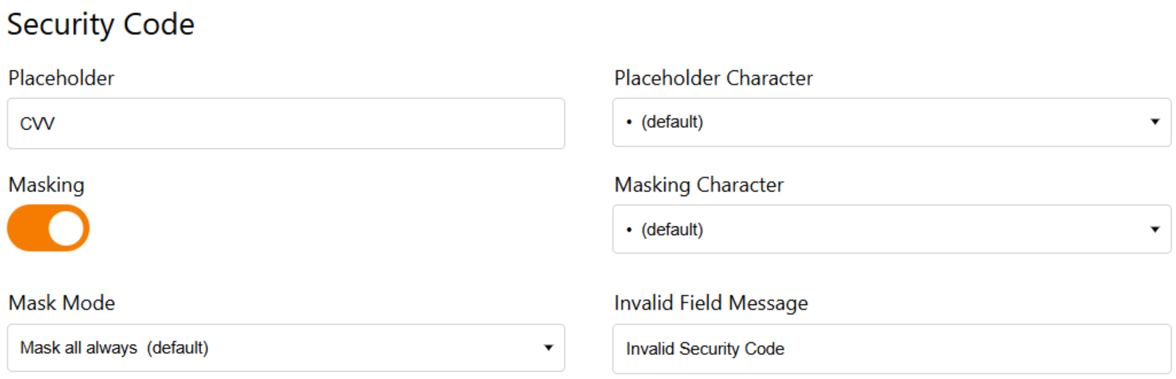
Invalid Field Message: Invalid Card Number



Name On Card

Placeholder: Name on Card

Invalid Field Message: Invalid Name on Card



Security Code

Placeholder: CVV

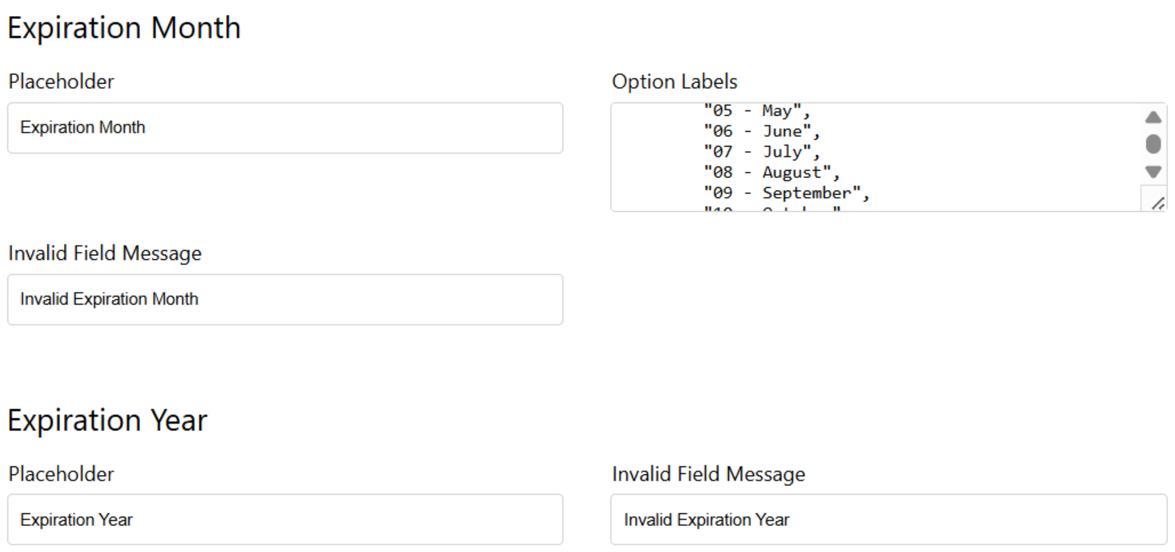
Placeholder Character: • (default)

Masking: ●

Masking Character: • (default)

Mask Mode: Mask all always (default)

Invalid Field Message: Invalid Security Code



Expiration Month

Placeholder

Expiration Month

Option Labels

```
        "05 - May",
        "06 - June",
        "07 - July",
        "08 - August",
        "09 - September",
```

Invalid Field Message

Invalid Expiration Month

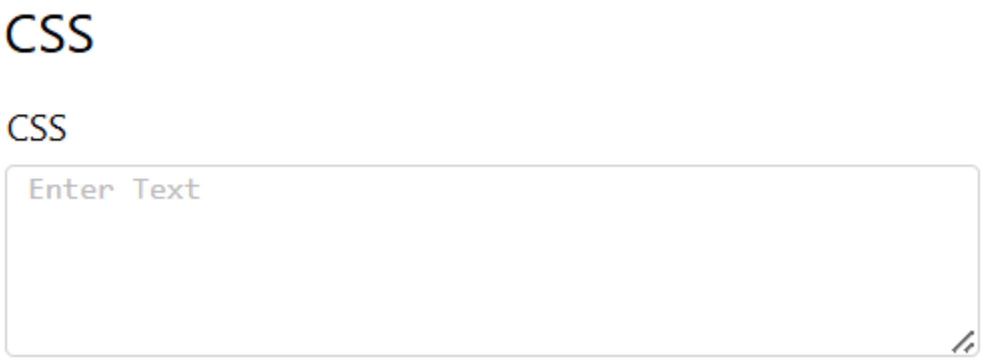
Expiration Year

Placeholder

Expiration Year

Invalid Field Message

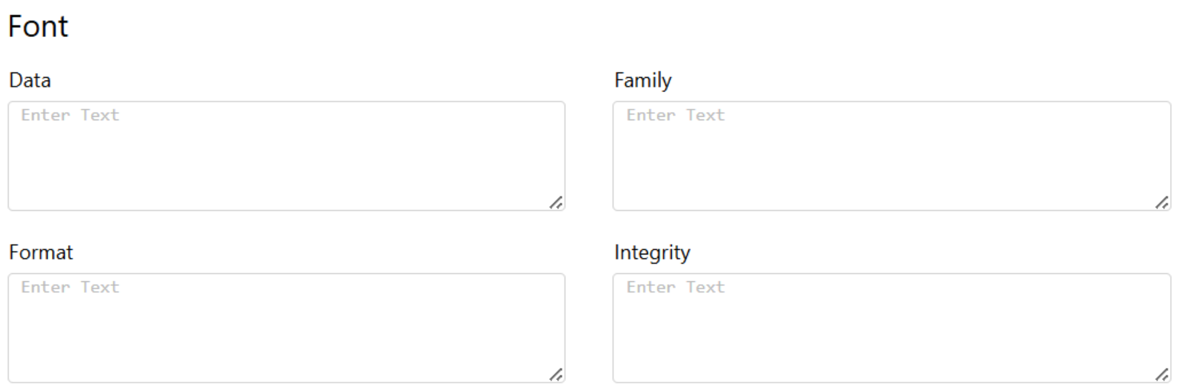
Invalid Expiration Year



CSS

CSS

Enter Text



Font

Data

Enter Text

Family

Enter Text

Format

Enter Text

Integrity

Enter Text

Default Customization

For convenience, we have provided default customization. The intention is to provide an example of what can be done, and merchants can implement according to their brand requirements.

CSS values passed into the corresponding Form CSS fields will style inputs embedded within the secure card capture iFrames. To style surrounding DOM elements within the payment form template, override styles in the following stylesheets:

- cartridge/static/default/css/fiservScc.css

Operations

Timeout Handling

The Fiserv Salesforce Extension has built-in management to handle timeout and latency issues between the extension (client) and Fiserv (server)-- covering a range of scenarios:

- Issues with requests between the client and server
- Issues with responses between the server and client
- Issues encountered during server-side processing

Within the merchant's MID configuration, a custom server side timeout less than what is set in the Salesforce plugin config is required. Please consult with your relationship manager / account manager to ensure this is set up properly.

To provide visibility on how timeout scenarios have been addressed, we employ the following steps.

> Step 1) Idempotency. If a transaction initiated by the Salesforce Extension (client) reaches the timeout, the extension will automatically retry the transaction leveraging Commerce Hub's idempotency feature.
>
> - If successful, the client will receive the proper result of the payment request and the customary workflow will resume.
> - If unsuccessful, perform Step 2.
>
> Step 2) If idempotency is unsuccessful, the Salesforce Extension (client) will perform a Transaction Inquiry request in an attempt to confirm the status of the original transaction.
>
> - If successful, the client will be returned the status and confirmation of the original payment request. Merchant workflow will resume.
> - If unsuccessful, perform Step 3.
>
> Step 3) If the Transaction Inquiry is unsuccessful, the Salesforce Extension then goes into recovery mode by doing the following:
>
> - For authorization requests, we will Cancel the transaction to release any funds that were potentially reserved on the consumer's form of payment.
> - For sale transactions, it is difficult for us to reverse as we do not want to attempt to reverse any money moving events. We suggest merchant operations investigate and take appropriate action with their consumers.

All timeout management steps above are logged and available for review. We suggest that there should be monitoring of these events by merchant operations.

Transaction Logging

System and transaction logs are critical for investigation and troubleshooting. Fiserv endeavors to provide operators with relevant and easy-to-use tools.

Transaction Log Enablement- Logging is automatically enabled defaulted to L2. This is so that a user can see both successful and errored transactions out-of-the-box.

Transaction Log Timezone – The default timezone for transaction logging is UTC.

Transaction Log Location – Logs are located by default within the Salesforce Business Manager. Fiserv Payments transaction logs can be found at:

> Administration > site development > development setup > log files

Logging Levels – The available logging levels have been designed to provide an increasing level of verbosity to assist operators in research and troubleshooting. There are a total of three logging levels.

> L1 represents the base log level that tracks the active process within the plugin.
>
> L2 reports errors. It will include both error messages and stack traces.
>
> L3 includes the most verbose information that includes information of objects created within the plugin throughout the transaction lifecycle.

At installation, logging is automatically set at L2, but can be changed without having to restart the Salesforce Commerce Cloud instance.

When the logging level is increased (e.g. from L2 to L3), lower-level logs will still be enabled and present. If a log level is decreased (e.g. from L3 to L1), log information appropriate to the new log level will be written.

L1 Log Example:

```
[2025-06-26 13:40:32.176 GMT] INFO PipelineCallServlet|1766663345|Sites-RefArch-Site|Fiserv-Credentials|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Intitating Credentials Request
[2025-06-26 13:40:33.010 GMT] INFO PipelineCallServlet|1766663345|Sites-RefArch-Site|Fiserv-Credentials|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Credentials Request Successful
[2025-06-26 13:40:34.148 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Initiating Early Tokenization call
[2025-06-26 13:40:34.149 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Initiating Card Tokenization
[2025-06-26 13:40:34.150 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Sending request to Commerce Hub
[2025-06-26 13:40:34.988 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Response received from Commerce Hub
[2025-06-26 13:40:34.990 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Duplicate card not stored.
```

```
[2025-06-26 13:40:34.991 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Tokenization Request Successful
[2025-06-26 13:40:34.991 GMT] INFO PipelineCallServlet|1061269059|Sites-RefArch-Site|PaymentInstruments-EarlyTokenization|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] Transaction ID: a82df7331d2047c1986fad876b7482a8
[2025-06-26 13:40:39.243 GMT] INFO PipelineCallServlet|8168756|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] [Order:00004524] Initiating Token Sale Transaction
[2025-06-26 13:40:39.245 GMT] INFO PipelineCallServlet|8168756|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] [Order:00004524] Sending request to Commerce Hub
[2025-06-26 13:40:41.377 GMT] INFO PipelineCallServlet|8168756|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] [Order:00004524] Response received from Commerce Hub
[2025-06-26 13:40:41.382 GMT] INFO PipelineCallServlet|8168756|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] [Order:00004524] Payment Token Sale Transaction Successful
[2025-06-26 13:40:41.382 GMT] INFO PipelineCallServlet|8168756|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|WNODtFSYaN custom.Fiserv []  [L1] [Order:00004524] Transaction ID: ed4ae32bb54541978520c09d10e944f8
```

L2 Log Example:

```
[2025-06-26 13:19:20.818 GMT] ERROR PipelineCallServlet|102635271|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|lw04pu4Wny custom.Fiserv [] [L2] [Order:00004518] Network call failed with error: 504
[2025-06-26 13:19:20.830 GMT] ERROR PipelineCallServlet|102635271|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|lw04pu4Wny custom.Fiserv [] [L2] [Order:00004518] Response Status: ERROR | Response Text: Transaction timeout
[2025-06-26 13:19:20.830 GMT] ERROR PipelineCallServlet|102635271|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|lw04pu4Wny custom.Fiserv [] [L2] [Order:00004518] Fiserv: Error: CommercehubCharges service call error code 504 Error => ResponseStatus: ERROR | ResponseErrorText: Transaction timeout | ResponseText: Gateway Timeout in fiserv_commercehub/cartridge/scripts/utils/commercehubServices.js:132
[2025-06-26 13:19:20.847 GMT] ERROR PipelineCallServlet|102635271|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|lw04pu4Wny custom.Fiserv [] [L2] [Order:00004518] An error occurred during payment processing. Checking for gift card transactions to reverse
[2025-06-26 13:19:20.848 GMT] ERROR PipelineCallServlet|102635271|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|lw04pu4Wny custom.Fiserv [] [L2] [Order:00004518] No gift transactions applied. Continuing...
```

L3 Log Example:

```
[2025-06-17 10:07:17.184 GMT] DEBUG PipelineCallServlet|1876092002|Sites-RefArch-Site|CheckoutServices-PlaceOrder|PipelineCall|38k-1ezisu custom.Fiserv [] [L3] [Order:00004101] Response Body:
{
  "gatewayResponse": {
    "transactionType": "CHARGE",
    "transactionState": "AUTHORIZED",
    "transactionOrigin": "ECOM",
    "transactionProcessingDetails": {
      "orderId": "CHG01c36118450402d97dc73a592bb6a020cd",
      "transactionTimestamp": "2025-06-17T10:07:15.426035264Z",
      "apiTraceId": "072165ac177940f7b901502814ebf6c8",
      "clientRequestId": "1802377155801",
      "transactionId": "072165ac177940f7b901502814ebf6c8",
      "apiKey": "AkfAAfn6jtD0wNn5luIPHGftBSgPeW3B"
    }
  },
  "paymentReceipt": {
    "approvedAmount": {
      "total": 150.14,
      "currency": "USD"
    },
    "processorResponseDetails": {
      "approvalStatus": "APPROVED",
      "approvalCode": "OK6105",
      "referenceNumber": "502814ebf6c8",
      "processor": "FISERV",
      "host": "NASHVILLE",
      "networkInternationalId": "0001",
      "responseCode": "000",
      "responseMessage": "Approved",
      "hostResponseCode": "00",
      "hostResponseMessage": "APPROVAL",
      "additionalInfo": [
        {
          "name": "COUNTRY_CODE",
          "value": "USA"
        },
      {
        "name": "CARD_PRODUCT_ID",
        "value": "X"
      },
      {
        "name": "DETAILED_PRODUCT_ID",
        "value": "H"
      }
    ],
    "bankAssociationDetails": {
      "associationResponseCode": "V000",
      "avsSecurityCodeResponse": {
        "streetMatch": "MATCHED",
        "postalCodeMatch": "MATCHED",
        "securityCodeMatch": "NOT_CHECKED",
        "association": {
          "avsCode": "Y",
          "securityCodeResponse": "X"
        }
      }
    }
  }
},
"billingAddress": {
  "firstName": "fname",
  "lastName": "lname",
  "address": {
    "street": "address1",
    "city": "Alpharetta",
    "stateOrProvince": "AL",
    "postalCode": "12345",
    "country": "US"
  },
  "phone": {
    "phoneNumber": "3232323232"
  }
},
"transactionDetails": {
  "captureFlag": false,
  "transactionCaptureType": "gateway",
  "merchantTransactionId": "eaef698c9b2d8b6800d20b3ca8",
  "merchantOrderId": "00004101",
  "createToken": true,
  "accountVerification": false,
  "partialApproval": true,
  "retrievalReferenceNumber": "502814ebf6c8",
  "processingCode": "000000"
},
"transactionInteraction": {
  "origin": "ECOM",
  "eciIndicator": "CHANNEL_ENCRYPTED",
  "posEntryMode": "MANUAL",
  "posConditionCode": "CARD_NOT_PRESENT_ECOM",
  "authorizationCharacteristicsIndicator": "W",
  "hostPosEntryMode": "010",
  "hostPosConditionCode": "59",
  "additionalPosInformation": {
    "stan": "144694",
    "posFeatures": {
      "pinAuthenticationCapability": "UNSPECIFIED",
      "terminalEntryCapability": "UNSPECIFIED"
    }
  }
},
"merchantDetails": {
  "tokenType": "EB03",
  "terminalId": "10000001",
  "merchantId": "100266000000162",
  "merchantPartner": {
    "id": "CSC909",
    "legacyTppId": "CSC909",
    "type": "PLUGIN",
    "name": "Salesforce",
    "productName": "Salesforce B2C Commerce",
    "versionNumber": "1.0.0",
    "integrator": "9876543212345678"
  }
},
"dynamicDescriptors": {
  "mcc": "7375",
  "merchantName": "Bomgaars Cert 2",
  "address": {
    "postalCode": "30040"
  }
},
"additionalDataCommon": {
  "marketSpecificDataIndicator": " ",
  "customFields": [
    {
      "key": "ECI",
      "value": "07"
    }
  ]
},
"networkDetails": {
  "network": {
    "network": "Visa",
    "accountFundingSource": "C",
    "responseSourceReasonCode": "B"
  },
  "networkResponseCode": "00",
  "cardLevelResultCode": "H",
  "validationCode": "G733",
  "transactionIdentifier": "015168127645043"
},
"cardDetails": {
  "binSource": "FISERV",
  "recordType": "DETAIL",
  "lowBin": "4111111111",
  "highBin": "4111111111",
  "binLength": "10",
  "binDetailPan": "16",
  "countryCode": "USA",
  "detailedCardProduct": "VISA",
  "detailedCardIndicator": "DEBIT",
  "pinSignatureCapability": "SIGNATURE",
  "issuerUpdateYear": "22",
  "issuerUpdateMonth": "04",
  "issuerUpdateDay": "22",
  "regulatorIndicator": "REGULATED",
  "cardClass": "CONSUMER",
  "anonymousPrepaidIndicator": "NON_ANONYMOUS",
  "productId": "MCS",
  "accountFundSource": "DEBIT",
  "panLengthMin": "16",
  "panLengthMax": "16",
  "moneySendIndicator": "UNKNOWN"
},
"source": {
  "sourceType": "PaymentCard",
  "card": {
    "nameOnCard": "TestOrder",
    "expirationMonth": "04",
    "expirationYear": "2033",
    "bin": "411111",
    "last4": "1111",
    "scheme": "VISA"
  }
},
"paymentTokens": [
    {
      "tokenData": "1079847268701111",
      "tokenSource": "TRANSARMOR",
      "tokenResponseCode": "000",
      "tokenResponseDescription": "SUCCESS"
    }
  ]
}
```

Known Issues / Questions

- Plugin has not been tested yet on mobile, therefore there may be some stylistic issues on mobile devices.
- During the checkout process, if the customer unknowingly adds the same gift card twice and proceeds to finalize the transaction, the system halts the checkout and displays a generic error message regarding “high order volume or temporary connection errors”. Instead, there should be a more descriptive error message relating to a duplicate gift card entry.

Feedback

TBD