

# Adobe Extension by Carat from Fiserv Payments

V1.0.0

## Introduction

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

[Commerce Hub](#) is Fiserv's modernized payment technology stack 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.

Please follow this link to documentation for Fiserv's in-house built Adobe Commerce extension.  
[ Link TBD ]

## Adobe Extension Overview

For the first time, an Adobe 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, but then may require fraud tools to combat risk and chargebacks. 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 Adobe 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.

## 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 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 to be subsequently followed by one or more captures.
  - Supports void of a valid authorized transactions in cases where an order is canceled before fulfillment.
  - Supports full and multiple partial captures to manage complete fulfillment of an order, partial fulfillment, and split shipment
- Supports full and multiple partial refunds for returns of full and partial orders.
- 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, Valuelink gift cards, and Adobe store credit. Only one of each payment method can be applied towards a given order.

#### *Valuelink 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 and/or store credit

Supports a consumer to present a gift card to purchase goods and services as a “sale”.

- \*\*NOTE: A gift card sale cannot be refunded, but only voided at this time.
- Supports a consumer to present a gift card to purchase goods and services as an “authorization only” flow to be subsequently followed by only one capture.
- Supports voids for gift card authorizations.
  - 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, Valuelink gift card and store credit.
  - A maximum of one of each payment method can be applied for a given order at this time.
  - 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
- Valuelink 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.
- Void  
This function voids (e.g. cancels) valid authorizations that have been secured on a payment method. This returns the reserved amount back to the consumer payment method. This can be executed so long as a capture has not been initiated. This is typically used for scenarios where an order needs to be canceled.
- Capture
  - Full
  - Multiple Partial (not supported by gift cards)
- Refund
  - Full
  - Multiple Partial

**\*\*NOTE.** Valuelink refunds are not supported. Most merchants have a refund policy that returns funds to either store credit or a different gift card.

- 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.
- Valuelink 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 three payment methods – a credit/debit card, a Valuelink gift card, and Adobe's store credit.
  - **\*\*NOTE.** At the moment the extension only allows the use of only one of each payment method at a time.

## 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.

## Adobe Commerce Platform Compatibility

### Edition:

- Adobe Commerce (cloud)
- Adobe Commerce (on-prem)
- Magento Open Source

### Adobe Commerce Store Version:

- >=2.4 (Adobe Commerce (cloud))

### [Fiserv Checkout SDK](#)

- Version 3.1.9

### Browsers 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 Fiserv's Developer Studio ([Dev Studio-- https://developer.fiserv.com/](https://developer.fiserv.com/))
- 2) Create an account / login
- 3) Create a workspace
  - a. Create a Merchant ID and Terminal 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.

Add Merchant ID

View

Add API key

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 Adobe extension by Carat from Fiserv Payments
- 4) Configure Adobe 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. Admin interface. Integration into Adobe admin 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

There are two options to access and download the Adobe extension by Carat from Fiserv Payments.

### Option 1: Install via Composer (recommended)

Add Fiserv's repository using the following command line:

```
config repositories.fiserv_payments git https://
github.com/Fiserv/adobe-commerce-plugin.git
```

Require Carat from Fiserv Payments

```
composer require fiserv/payments:dev-master#v0.0.5
```

### Option 2: Install compressed library (e.g. .tar / .zip file)

Extract compressed plugin to the following directory:

```
<Adobe Commerce Root Directory>/app/code/Fiserv
```

Upon downloading with either Option 1 or 2, please follow the below steps to enable the extension:

- 1) Enable Carat from Fiserv Payments module  

```
php bin/magento module:enable Fiserv_Payments --clear-static-content
```
- 2) Update modules  

```
php bin/magento setup:upgrade
```
- 3) Deploy static content  

```
php bin/magento setup:static-content:deploy -f
```
- 4) Clear cache  

```
php bin/magento c:c
php bin/magento c:f
```
- 5) For security reasons, Fiserv requires that all domains for development, UAT, production, etc.) must be whitelisted. Whitelisting can be facilitated through your sales or account representative.

**\*\*NOTE:** Valuelink has been added as a form of payment. To support, the installation will include the following schema updates.

- \* New table: valuelink\_transaction
- \* Altered table: quote
  - \* Added column valuelink\_cards

- \* Added column valuelink\_cards\_amount
  - \* Added column base\_valuelink\_cards\_amount
- \* Altered table: quote\_address
  - \* Added column valuelink\_cards
  - \* Added column valuelink\_cards\_amount
  - \* Added column base\_valuelink\_cards\_amount

## Step 2: Carat from Fiserv Payments Configuration

**!!! 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 Adobe extension from Carat 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 Carat from Fiserv Payments configuration. Within the Adobe Commerce Administrative panel, navigate to:

Stores > Configuration > Sales > Payment Methods > Carat from Fiserv Payments

### Navigate to Stores

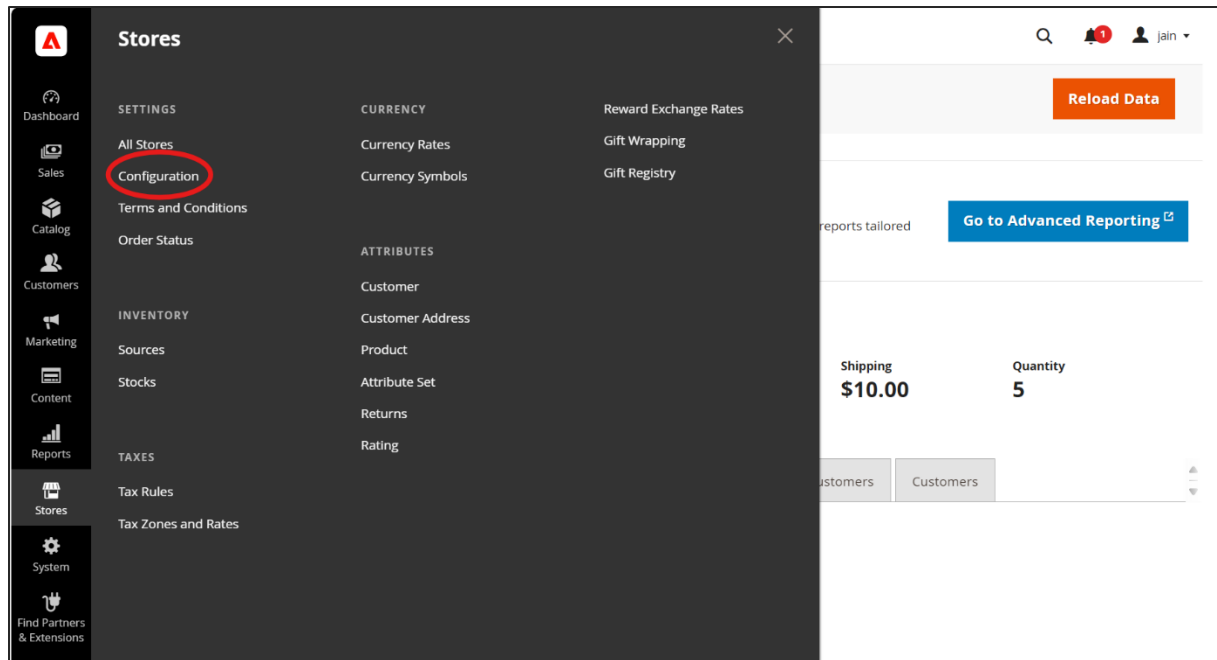
**Stores** > Configuration > Sales > Payment Methods > Carat from Fiserv Payments

The screenshot shows the Adobe Commerce Dashboard. On the left sidebar, the 'Stores' menu item is highlighted with a red circle. The main content area displays various business metrics and reports. The 'Last Orders' table is visible at the bottom.

Customer	Items	Total
test jainBomga	1	\$24.00
Test Tester	2	\$19.99
Test Tester	2	\$17.98
JainCH BomgaarsLTra	1	\$0.00
commhubBomGars JainCH	2	\$7.99

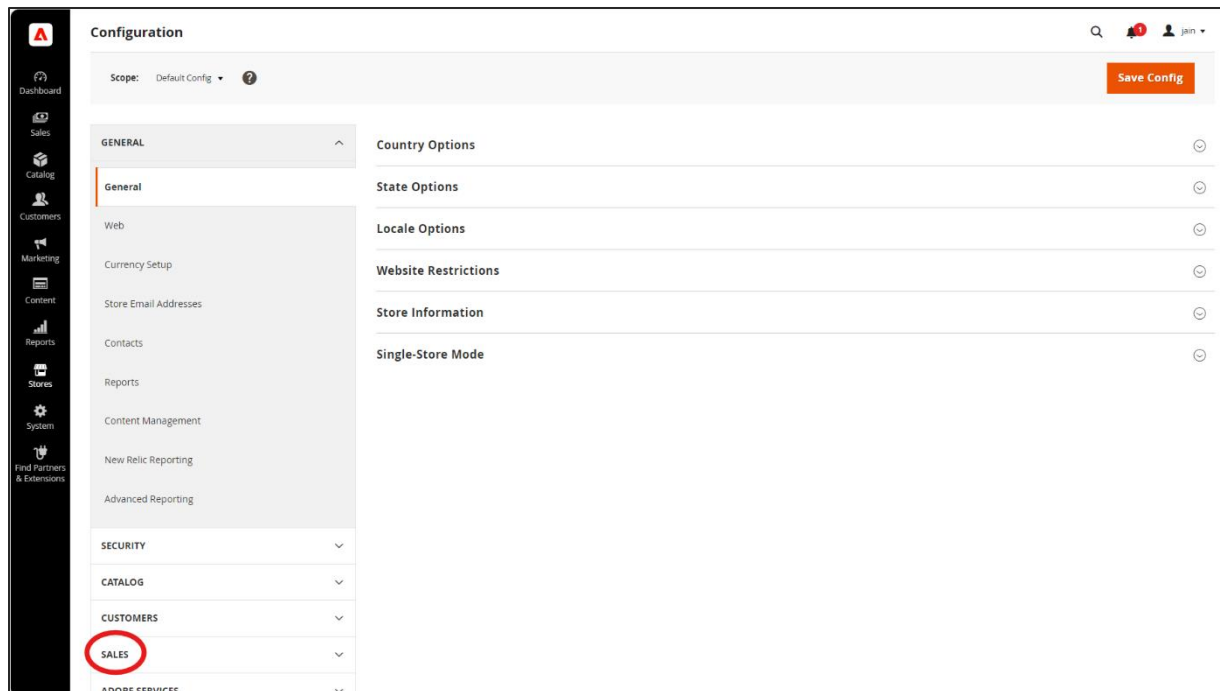
## Navigate to Configuration

Stores > **Configuration** > Sales > Payment Methods > Carat from Fiserv Payments



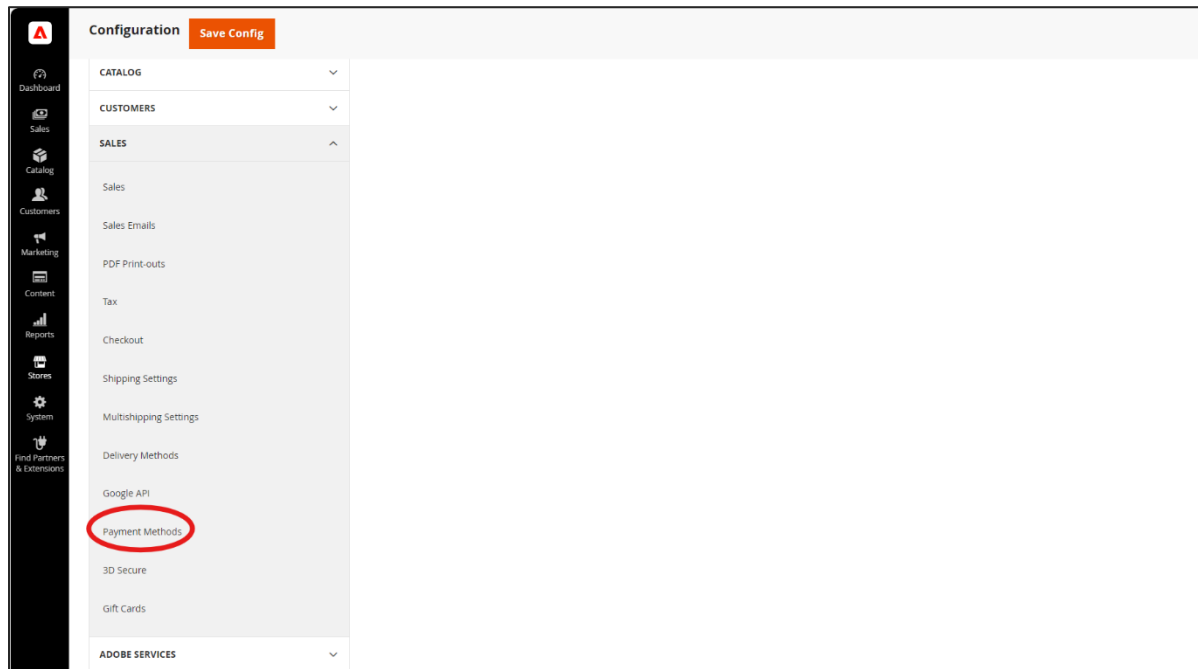
## Navigate to Sales

Stores > Configuration > **Sales** > Payment Methods > Carat from Fiserv Payments



## Navigate to Payment Methods

Stores > Configuration > Sales > **Payment Methods** > Carat from Fiserv Payments



## Navigate to Fiserv Payments

Stores > Configuration > Sales > Payment Methods > **Carat from Fiserv Payments**

**\*\*NOTE: Carat from Fiserv Payments will be listed under "OTHER PAYMENT METHODS"**

Dashboard

Sales

Catalog

Customers

Marketing

Content

Reports

Stores

System

Find Partners & Extensions

Configuration

Save Config

Multishipping Settings

Delivery Methods

Google API

Payment Methods

3D Secure

Gift Cards

ADOBE SERVICES

SERVICES

ADVANCED

ADDITIONAL PAYMENT SOLUTIONS: ▼

OTHER PAYMENT METHODS: ▲

Zero Subtotal Checkout

Check / Money Order

Bank Transfer Payment

Cash On Delivery Payment

Purchase Order

Carat from Fiserv Payments

Carat is the global commerce platform from Fiserv that orchestrates payments and experiences for the world's largest businesses. Commerce Hub is Fiserv's modernized payment technology stack that facilitates the enablement of payments and value added services for omnichannel. A connection to Commerce Hub will afford any merchant a single solution to meet their growing commerce needs.

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

Commerce Hub Gateway General Settings

Payment Acceptance Settings

Credit/Debit Cards

Valuelink Gift Cards

Payment Form Customization

Copyright © 2024 Adobe. All rights reserved.

Adobe Commerce ver. 2.4.7-p1

Privacy Policy | Report an Issue

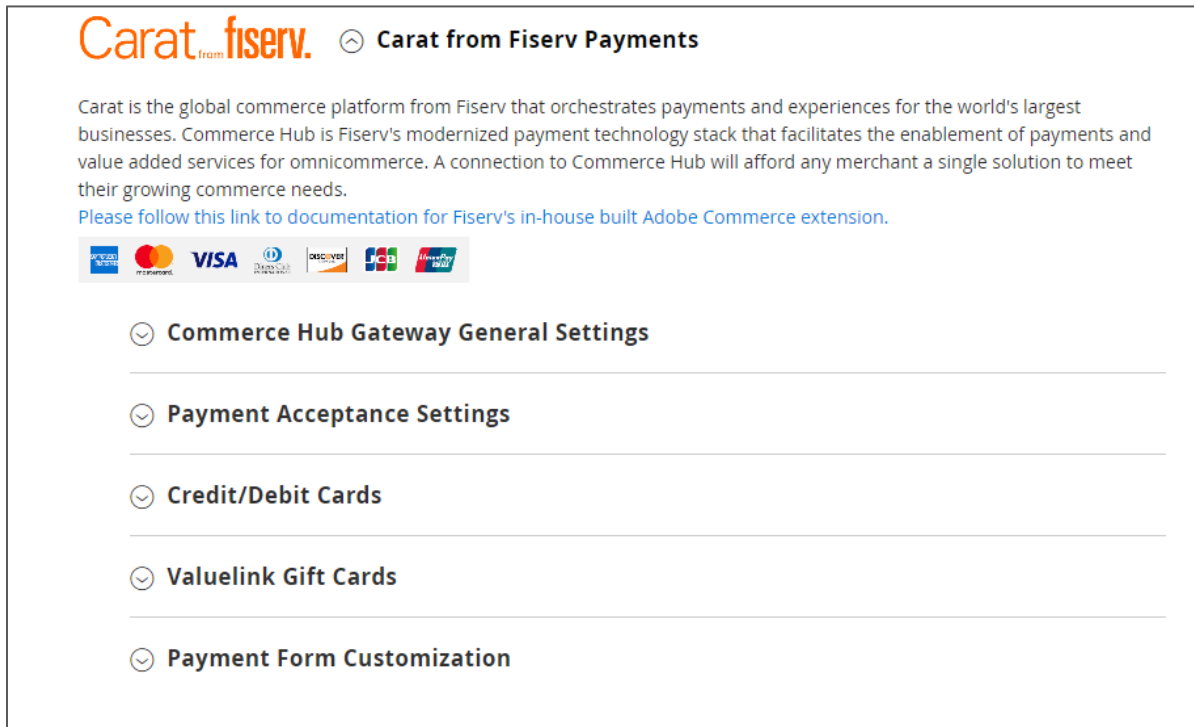


## Expand Commerce Hub Gateway Settings

Stores > Configuration > Sales > Payment Methods > Carat from Fiserv Payments >

### Commerce Hub Gateway General Settings

Expand the “Carat from Fiserv Payments” payment method to expose “Commerce Hub Gateway General Settings”



Upon expansion, a list of configuration areas will be revealed. All configurations will be done on this page. The configuration section has been categorized the following:

- Commerce Hub General Settings
- Payment Acceptance Settings
- Credit/Debit Cards
- Valuelink Gift Cards
- Payment form Customization

Details of these categories are explained in Step 3.

## Saving Configurations & Cache Management

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

The screenshot displays the 'Configuration' page in the Fiserv Carat interface. On the left is a vertical sidebar with navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main header area contains the 'Configuration' tab and a 'Save Config' button, which is circled in red. A red arrow points from the 'Save Config' button down to the 'ADVANCED' section of the configuration menu. The configuration menu has two sections: 'SERVICES' and 'ADVANCED'. The 'ADVANCED' section is expanded, showing various settings. The main content area is titled 'Carat from Fiserv Payments' and includes a description of the platform, a link to documentation, and logos for supported payment methods: American Express, Mastercard, Visa, Diners Club International, Discover, JCB, and UnionPay. Below the logos are two main sections: 'Commerce Hub Gateway General Settings' and 'Payment Acceptance Settings'. The 'Commerce Hub Gateway General Settings' section includes fields for 'Enable Fiserv Commerce Hub Gateway' (set to 'Yes'), 'Merchant ID', 'Terminal ID', 'API Key', 'API Secret', 'API Environment' (set to 'CERT'), 'Debug' (set to 'No'), and 'Logging Level' (set to 'L3 --- In depth, detailed logs (Development)'). The 'Payment Acceptance Settings' section includes fields for 'Charge Type' (set to 'Authorize'), 'Accepted Currency' (set to 'US Dollar'), and 'Accept Payments From All Countries' (set to 'All Allowed Countries'). A note below the 'Accepted Currency' field states: 'This version supports USD only. Additional currencies will be added in the future.'

**Configuration** **Save Config**

**SERVICES** **ADVANCED**

**Carat from Fiserv Payments**

Carat is the global commerce platform from Fiserv that orchestrates payments and experiences for the world's largest businesses. Commerce Hub is Fiserv's modernized payment technology stack that facilitates the enablement of payments and value added services for omnicommerce. A connection to Commerce Hub will afford any merchant a single solution to meet their growing commerce needs.  
[Please follow this link to documentation for Fiserv's in-house built Adobe Commerce extension.](#)

**Commerce Hub Gateway General Settings**

Enable Fiserv Commerce Hub Gateway (website) Yes

Merchant ID (website) \*\*\*\*\*

Terminal ID (website) \*\*\*\*\*

API Key (website) \*\*\*\*\*

API Secret (website) \*\*\*\*\*

API Environment (website) CERT

Debug (website) No

Logging Level (website) L3 --- In depth, detailed logs (Development)

**Payment Acceptance Settings**

Charge Type (website) Authorize

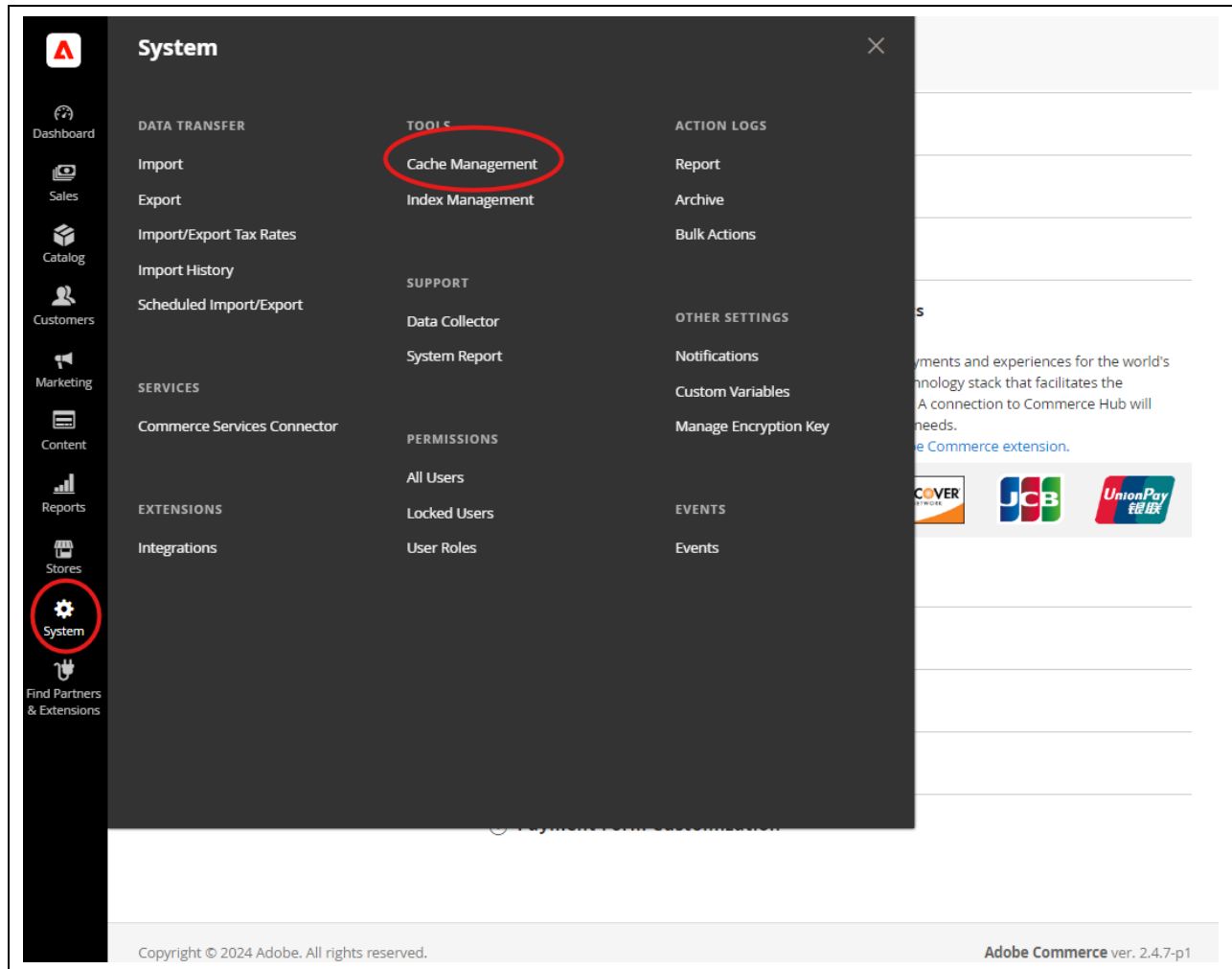
Accepted Currency (website) US Dollar  
This version supports USD only. Additional currencies will be added in the future.

Accept Payments From All Countries (website) All Allowed Countries

Upon saving, Adobe Commerce requires one to refresh the configuration by flushing the cache. To do this, you must do the following:

### Navigate to Cache Management

System > **Cache Management**



Once within the Cache Management console, choose “Page Cache” and then “Flush Magento Cache”.

The screenshot shows the 'Cache Management' interface. On the left is a sidebar with navigation links: Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main header includes a search icon, a notification bell with '1', and a user profile 'jain'. Below the header, there are two buttons: 'Flush Cache Storage' and 'Flush Magento Cache' (highlighted with a red circle). A status bar indicates '18 records found (1 selected)'. The table below lists various cache types, with 'Page Cache' selected (checkbox checked and circled in red).

<input type="checkbox"/>	Cache Type	Description	Tags	Status
<input type="checkbox"/>	Configuration	Various XML configurations that were collected across modules and merged	CONFIG	ENABLED
<input type="checkbox"/>	Layouts	Layout building instructions	LAYOUT_GENERAL_CACHE_TAG	ENABLED
<input type="checkbox"/>	Blocks HTML output	Page blocks HTML	BLOCK_HTML	ENABLED
<input type="checkbox"/>	Collections Data	Collection data files	COLLECTION_DATA	ENABLED
<input type="checkbox"/>	Reflection Data	API interfaces reflection data	REFLECTION	ENABLED
<input type="checkbox"/>	Database DDL operations	Results of DDL queries, such as describing tables or indexes	DB_DDL	ENABLED
<input type="checkbox"/>	Compiled Config	Compilation configuration	COMPILED_CONFIG	ENABLED
<input type="checkbox"/>	EAV types and attributes	Entity types declaration cache	EAV	ENABLED
<input type="checkbox"/>	Webhooks Response Cache	Caches responses to webhook requests	WEBHOOKS_RESPONSE	ENABLED
<input type="checkbox"/>	Customer Notification	Customer Notification	CUSTOMER_NOTIFICATION	ENABLED
<input type="checkbox"/>	Integrations Configuration	Integration configuration file	INTEGRATION	ENABLED
<input type="checkbox"/>	Integrations API Configuration	Integrations API configuration file	INTEGRATION_API_CONFIG	ENABLED
<input type="checkbox"/>	GraphQL Query Resolver Results	Results from resolvers in GraphQL queries	GRAPHQL_QUERY_RESOLVER_RESULT	ENABLED
<input type="checkbox"/>	Admin UI SDK Cache	Caches Apps customizations to the Admin Panel	ADMIN_UI_SDK	ENABLED
<input checked="" type="checkbox"/>	Page Cache	Full page caching	FPC	ENABLED
<input type="checkbox"/>	Target Rule	Target Rule Index	TARGET_RULE	ENABLED
<input type="checkbox"/>	Web Services Configuration	REST and SOAP configurations, generated WSDL file	WEBSERVICE	ENABLED


**\*\*NOTE:** Each time a configuration setting is saved, the cache must be flushed for the changes to take effect.

### Step 3: Configuration Instructions for Carat from Fiserv Payments

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.

 **Commerce Hub Gateway General Settings**

Enable Fiserv Commerce Hub Gateway  
[website]

Yes

Merchant ID  
[website]

.....

Terminal ID  
[website]

.....

API Key  
[website]

.....

API Secret  
[website]

.....

API Environment  
[website]

CERT

Debug  
[website]

No

Logging Level  
[website]

L1 --- Basic information

L1 --- Basic information

L2 --- Error reporting

L3 --- In depth, detailed logs (Development)


Configuration Field	Value	Description
Enable Fiserv Commerce Hub Gateway	Boolean: Yes / No	Setting this to Yes enables the use of the extension.
Merchant ID	String	Provided through Dev Studio.
Terminal ID	String	Provided through Dev Studio.
API Key	String	Provided through Dev Studio.

API Secret	String	<p>Provided through Dev Studio.</p> <p><b>**NOTE.</b> 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.</p>
API Environment	<p>Values:</p> <ul style="list-style-type: none"> <li>• CERT</li> <li>• PROD</li> </ul>	<p>Fiserv supports two environments – CERT and PROD.</p> <p><b><u>CERT</u></b> 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.</p> <p><b><u>PROD</u></b> 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.</p>
Debug	Boolean: Yes / No	This configuration previously enabled and disabled logging of the plugin. This has since been deprecated and replaced by the Logging Level configuration.
Logging Level	<p>Values:</p> <ul style="list-style-type: none"> <li>• L1 – Basic information</li> <li>• L2 – Error reporting</li> <li>• L3 – In depth, detailed logs (Development)</li> </ul>	See the <a href="#">Transaction Logging</a> section to learn more.



## Payment Acceptance Settings

This area determines the general payment acceptance in the extension.

 **Payment Acceptance Settings**

Charge Type  
[website]

Authorize

Accepted Currency  
[website]

US Dollar

This version supports USD only. Additional currencies will be added in the future.

Accept Payments From All Countries  
[website]

All Allowed Countries

Accept Payments From Select Countries  
[website]

Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Andorra

Angola

Anguilla

Antarctica

Antigua & Barbuda

Configuration Field	Value	Description
Charge Type	Values: <ul style="list-style-type: none"><li>Sale (Authorization + Capture)</li><li>Authorize</li></ul>	<p><b><u>Sale (Authorize + Capture)</u></b> will fully charge the consumer at the time of checkout.</p> <p><b><u>Authorize</u></b> requires a subsequent capture request. This charge type will place a hold / reserve on funds.</p>




Accepted Currency	US Dollar	<p>US Dollar (USD) Only</p> <p><b>**NOTE.</b> In this version of the plugin, only US Dollar (USD) is enabled. More currencies will be available over time.</p>
Accept Payments From All Countries	<p>Values:</p> <ul style="list-style-type: none"> <li>• All Allowed Countries</li> <li>• Specific Countries</li> </ul>	<p>This configuration determines from which countries consumers can purchase from the merchant website.</p> <p>This is typically based on merchant business rules and their operational ability to service consumers.</p> <p>If "<b><u>All Allowed Countries</u></b>" is chosen, a merchant can accept payments from all countries supported by Fiserv. All Allowed Countries list is the list countries that Fiserv can support-- which considers geo-political and sanction mandates.</p> <p>If "<b><u>Specific Countries</u></b>" is chosen the following configuration field "<b>Accept Payments From Select Countries</b>" will be enabled. Else, it will be grayed out and immutable.</p>
Accept Payments From Select Countries	Dropdown List	<p>This configuration is enabled when "<b><u>Accept Payments from Select Countries</u></b>" is chosen as the value of</p>

		<p>the previous field. One to many may be selected.</p> <p><u>At least ONE country</u> must be chosen, else the integration will not work.</p>
--	--	--

Credit/Debit Cards

This section defines payment card specific settings.

 **Credit/Debit Cards**

Payment Method Title  
[website]

Fiserv CommerceHub (Payment Card)

Allow Customers To Store (Tokenize)  
Payment Methods  
[website]

Yes

Allow Customers To Manage Stored  
Payment Methods  
[website]

Yes

Configuration Field	Value	Description
Payment Method Title	String	This is the label shown to the consumer for the credit / debit card section of available payment methods.
Allow Customers To Store (Tokenize) Payment Methods	Boolean: Yes / No	<p>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.</p> <p>For future purchases when logged in, consumers can select a stored card on file rather than entering a new card.</p> <p><u>Yes</u>, means that a check box will be shown at checkout so that a consumer can opt to save their card details; <b>DEFAULTED to Yes</b></p> <p><u>No</u>, means that a check box will NOT be shown at checkout. A consumer card does not have the option of being saved.</p>
Allow Customers To Manage Stored Payment Methods	Boolean: Yes / No	This configuration is tied to the above value. If "Allow Customers To Store (Tokenize) Payment

		<p>Methods" = No, then this configuration field is not an option and will not even shown.</p> <p>If available,</p> <p><u>Yes</u>, 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).</p> <p><u>No</u>, means that a consumer CANNOT log into their merchant store account and save or delete a stored payment method.</p>
--	--	--

**Valuelink Gift Cards**

This section defines Valuelink gift card specific settings.

## Valuelink Gift Cards

Enable Valuelink Gift Cards  
[website]

Yes



Valuelink is a gift card solution native to Fiserv.  
Choose Yes to enable Valuelink as a payment  
method or leave No to disable Valuelink.

Payment Method Title  
[website]

Valuelink Gift Cards1

Charge Type  
[website]

Authorize



Max Number of Gift Cards  
Allowed per Order  
[global]

1



Number of gift cards currently restricted to 1 per  
order

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

Charge Type	Values: <ul style="list-style-type: none"> <li>• Sale (Authorization + Capture)</li> <li>• Authorize</li> </ul>	<p><b><u>Sale (Authorize + Capture)</u></b> will fully charge the consumer at the time of checkout.</p> <p><b><u>Authorize</u></b> requires a subsequent capture request. This charge type will place a hold / reserve on funds.</p>
Max Number of Gift Cards Allowed per Order	1	<p>Currently, only one gift card is allowed per order; however, the extension will soon support applying multiple gift cards per order.</p> <p>For now, it will be defaulted to “1” (immutable) and will be expanded in our next iteration.</p>

## 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 Adobe 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 Admin Order interface
- Within the consumer's account profile

We have provided the following customization areas for management control.

#### **Payment Form Customization**

[Payment Form Customization Documentation.](#)

##### **Customer Checkout Payment Form**

---

##### **Customer Checkout Valuelink Form**

---

##### **Admin Order Payment Form**

---

##### **Admin Order Valuelink Form**

---

##### **Tokenization 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 Adobe 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 Valuelink 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.

[JavaScript Example](#) **Variables**

```
placeholder
dynamicPlaceholderCharacter
enableFormatting
masking.character
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.

### Card Number Field


Placeholder  
[website]

Placeholder Character  
[website]  

Format Card Number  
[website]  

Mask Card Number  
[website]  

Masking Character  
[website]  

Mask Mode  
[website]  

Mask Length  
[website]

Invalid Field Message  
[website]

### Name on Card Field

Placeholder  
[website]

Invalid Field Message  
[website]

### Security Code Field

Placeholder  
[website]

Placeholder Character  
[website]  

Mask Security Code  
[website]  

Masking Character  
[website]  

Mask Length  
[website]  




Invalid Field Message  
[website]

### Expiration Month Field

Placeholder  
[website]

Option Labels  
[website] 

[  
January",  
February",  
"01 -  
"02 -  
"03 -



Invalid Field Message  
[website]

### Expiration Year Field

Placeholder  
[website]

Invalid Field Message  
[website]

## Form CSS

Checkout Form CSS  
[website]

```
{  
  "input, select" : {  
    "background-color" : "#ffffff"  
  },  
  "input::placeholder": {  
    "color": "#000"  
  }  
}
```

## Custom Font

All fields are required to apply a custom font to the payment form.

Font Data  
[website]

Font Family  
[website]

Font Format  
[website]

Integrity  
[website]

## 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:

1. Customer checkout form: `ch-form.css`
2. Admin order creation payment form: `ch-form-admin.css`
3. Stored payment accounts form: `ch-form-token.css`

## User Acceptance Testing (UAT)

Fiserv has tested and regressed our Adobe extension and feels that it is good practice to share relevant test scripts from a payment point of view with our extension users.

### Payment Test Scripts

Please refer to the accompanying test scripts provided along with this document. We strongly recommend that merchants use these to complement their own testing processes.

### End to End Testing

Check on Adobe Interface and Clientline (Commerce Hub's reporting interface).

# 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 Location** – Logs are located by default within the logging architecture of Adobe Commerce into a dedicated log file. Carat from Fiserv Payments transaction logs can be found at:

```
<Adobe Commerce Root Directory>/var/log/commerce_hub.log
```

**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 Adobe Commerce instance; however, the cache must be flushed for the new configurations to take effect.

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:

```
[2024-10-09T01:57:53.855833+00:00] CommerceHubLogger.INFO: [L1] Initiating Credentials Request [] []  
[2024-10-09T01:57:54.612885+00:00] CommerceHubLogger.INFO: [L1] Credentials request success [] []  
[2024-10-09T01:58:55.773643+00:00] CommerceHubLogger.INFO: [L1] Retrieving payment tokens [] []  
[2024-10-09T01:58:55.774269+00:00] CommerceHubLogger.INFO: [L1] Payment tokens retrieved [] []  
[2024-10-09T01:58:56.999798+00:00] CommerceHubLogger.INFO: [L1] starting execution [] []  
[2024-10-09T01:58:57.004390+00:00] CommerceHubLogger.INFO: [L1] Initiating Token Auth Transaction [] [] . . .  
[2024-10-09T10:19:39.443676+00:00] CommerceHubLogger.INFO: [L1] Transaction success [] []  
[2024-10-09T10:19:39.443726+00:00] CommerceHubLogger.INFO: [L1] Transaction ID:  
a91e532c15104b30b930f3e9caee4c12 [] []  
[2024-10-09T10:19:39.443726+00:00] CommerceHubLogger.INFO: [L1] Transaction ID:  
a91e532c15104b30b930f3e9caee4c12 [] []
```

### L2 Log Example:

```
[2024-10-09T12:58:59.249871+00:00] CommerceHubLogger.ERROR: [L1] Transaction failure. Commerce Hub response
returned with unsuccessful transaction state [] []
[2024-10-09T12:58:59.249922+00:00] CommerceHubLogger.ERROR: [L1] Transaction ID:
4ea07cf0e13e4daca5ad5264106fc2c [] []
[2024-10-09T12:58:59.249970+00:00] CommerceHubLogger.ERROR: [L2] Transaction state: DECLINED [] []
[2024-10-09T12:58:59.250014+00:00] CommerceHubLogger.ERROR: [L2] Response message: Declined [] []
[2024-10-09T12:58:59.250057+00:00] CommerceHubLogger.ERROR: [L2] Payment Source Type: PaymentToken [] []
[2024-10-09T12:58:59.285476+00:00] CommerceHubLogger.INFO: [L1] Processing ValueLink transactions. Total
transactions: 1 [] []
[2024-10-09T12:58:59.285558+00:00] CommerceHubLogger.INFO: [L1] Attempting to cancel ValueLink transaction ID:
0653fd60b4a14e56adb4ebff0c0a7de4 [] []
[2024-10-09T12:58:59.689786+00:00] CommerceHubLogger.INFO: [L1] Successfully canceled ValueLink transaction ID:
0653fd60b4a14e56adb4ebff0c0a7de4 [] []
[2024-10-09T12:58:59.689888+00:00] CommerceHubLogger.INFO: [L1] All ValueLink transactions were successfully
voided. [] []
L3 Log Example:
```

### L3 Log Example:

```
[2024-10-09T01:39:24.739971+00:00] CommerceHubLogger.INFO: [L3] Response Body:
```

```
{
  "gatewayResponse": {
    "transactionType": "CHARGE",
    "transactionState": "AUTHORIZED",
    "transactionOrigin": "ECOM",
    "transactionProcessingDetails": {
      "orderId": "CHG014d76b82860de691755fce82d46ace528",
      "transactionTimestamp":
        "2024-10-09T01:39:22.957077478Z",
      "apiTraceId": "50b5bfff73eeb441992a1ac7dca6a6f7e",
      "clientRequestId": "1728987324033",
      "transactionId": "50b5bfff73eeb441992a1ac7dca6a6f7e",
      "apiKey": "AkfAAfn6jtd0wNn5luIPHGftBSgPeW3B"
    }
  },
  "paymentReceipt": {
    "approvedAmount": {
      "total": 25,
      "currency": "USD"
    }
  },
  "processorResponseDetails": {
    "approvalStatus": "APPROVED",
    "approvalCode": "OK5783",
    "referenceNumber":
      "ac7dca6a6f7e", "processor":
      "FISERV", "host": "NASHVILLE",
    "networkRouted": "VISA",
    "networkInternationalId": "0001",
    "responseCode": "000",
    "responseMessage": "Approved",
    "hostResponseCode": "00",
    "hostResponseMessage": "APPROVAL",
    "additionalInfo": [
      { "name": "COUNTRY_CODE",
        "value": "USA"
      }
    ],
    {
      "name": "CARD_PRODUCT_ID",
      "value": "X"
    }
  },
  {
    },
    "networkDetails": {
      "network": {
        "network": "Visa" },
      "networkResponseCode": "00",
      "cardLevelResultCode": "H",
      "validationCode": "CN ",
      "transactionIdentifier": "014283786163273"
    }
  },
  "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": "PaymentToken",
    "tokenData": "1079847268701111",
    "card": {
      "expirationMonth": "01",
      "expirationYear": "2025",
      "bin": "411111",
      "last4": "1111",
      "scheme": "VISA"
    }
  },
  "accountType": "FPAN"
}
[] []
[2024-10-09T01:58:58.877531+00:00] CommerceHubLogger.INFO: [L1] Transaction success [] []
[2024-10-09T01:58:58.877581+00:00] CommerceHubLogger.INFO: [L1] Transaction ID:
e7f4a975ca574ba891b9473a5de19fb8 [] []

```

## Known Issues / Questions

- **Form Validations.**
  - For reasons flagged for ADA for those who are visually impaired, we have taken the liberty to produce a readable error that indicates a potential input error and asks for the information to be re-entered.
  - Iframe border validations can persist after reloading the form
  - Error banner for tokenization form can disappear after only a second of visibility
  - Iframe width on mobile screens can be incorrect
- **Decline Messages.** Where transactions result in soft and/or hard declines, the extension will report the reasons within the response object. It is fully available for a merchant to access, interpret, and render an appropriate message for the end user. We assume that this is the case to provide full control to a merchant.
- **Admin refresh issue.** When a merchant sales associate is entering an order on behalf of a merchant in the Adobe Commerce Admin interface, there is an edge case where if the page is refreshed, the form fields will disappear. With research, this is not within the scope of Fiserv's extension, but more so with the Adobe platform. We plan on reporting this to Adobe.

## Releases Notes

### Version 1.0.0

- Fixed.
  - Propagated gift card transactions to email invoices.
  - Miscellaneous cosmetic fixes.
- New Feature
  - Timeout reversal management.
- Completed regressions and compatibility tests for production

### Version 0.0.5

- Fixed. Updates and changes to .xml files and database schema for installation process.

### Version 0.0.4

- New Features.
  - Valuelink balance inquiry and redemption. Redemption includes both sale and authorization/capture functionality leveraging customizable field architecture.
  - Enhanced transaction logging with levels of verbosity.

- Split tender to include a payment card, valuelink card, and store credit.
- Upgrade to Checkout SDK 3.1.9 for better browser and form factor support.
- Fixed. Various functional and regression defects.

#### Version 0.0.3

- Fixed. Reported that API Key and Secret fields were applied a validation check that resulted in a false negative. Remediated by modifying the validation logic.
- Fixed. Logic was implemented to message an end user when input data was invalid – e.g. bad formatting. Reported that certain field input checks that triggered “invalid data” messages were not being cleared once and end user had corrected the input. Input values were correct, yet an invalid message was still shown. While it did not prevent transactions from executing, it did provide usability issues and confusion for end users. Remediated the validation logic to clear these messages once corrected.

#### Version 0.0.2

- Fixed. Reported that certain Adobe Commerce installations do not allow system.xml files with nested objects greater than three levels preventing the plugin from starting. Remediated by collapsing the nested objects.
- Fixed. Reported that the checkout place order button was not able to handle parallel modules. Issue uncovered when the Avatax and Fiserv modules could not be executed without forcing a second click by the user. Remediated by changing the order of operations of how a parallel module is called.
- Updated configurations. The Carat from Fiserv Payments configuration page has been revamped for better categorization and usability – including ability to set invalid field messages. New configurations are reflected in this document.

## Feedback

TBD

