

## Moneris Checkout

This article provides OPSCOM system admins with a comprehensive guide to setting up and configuring Moneris Checkout as a payment provider. Moneris Checkout is a full-service partner that allows you to support a wide range of payment options within OPSCOM.

---

### Setup and Configuration

Properly configuring Moneris Checkout involves two key stages: configuring the settings within your Moneris merchant account and then mirroring the appropriate settings within OPSCOM. **We recommend using the pre-auth settings in both Moneris and OPSCOM for the most reliable payment process.**

**See this gif image, which shows what setting to change on the Moneris Merchant Resource Center, in order to switch from purchase to pre-authorization. (shown in the image below, beside capture method)**

QA Merchant 5 - e-troud - store5

Admin ▾ Reports ▾ Terminal ▾ Receiving ▾ Vault ▾ Help ▾ Demo User ▾

# Moneris Checkout Configuration

[Back to Checkout Configurations](#) [Favourite](#)

### MOC-SPO

[Edit Name](#) | [Delete Profile](#)

**Checkout Type**

Use Moneris Checkout for the complete set of online payment forms.  
Select this option if you want to use Moneris forms for order summary, payment, shipping details, and order confirmation.

I have my custom order form and want to use Moneris simply for payment processing.  
Select this option if you want to embed Moneris payment details within your own order form.

[Multi-Currency](#)


### Checkout ID

chkt/WUNtrou5 [Save](#)

[Full Screen Preview](#)

Choose one of the following express checkout options or enter payment details manually.

**Payment Method**



Cardholder's Name

Card Number

MMYY  CVV

## Information Required for Setup

Type Name: JS: Moneris Checkout

Description: Inline Checkout with MonerisCheckout

Enabled For User Types: Settings

Enable for Text2ParkMe:

Enable for Permit Renewal Payments:

Enable for Guest Payments:

Can Change Type After Being Process?: This payment type cannot be changed

Prompt Information:
 

- Email Address
- Billing Details
- Email Address

Production Credentials
 

- Store ID: store5
- API Token: yesguy
- Checkout ID: chktK478Store5
- Hosted Token ID: htI8Q3NCRRXQXLC

Capture Method:
  Preauthorization  Purchase

Testing Credentials
 

- Store ID: store5
- API Token: yesguy
- Checkout ID: chktK478Store5
- Hosted Token ID: htI8Q3NCRRXQXLC

Other Settings:
  Hidden  Visible  Required

Update Payment Type

Some options on Moneris can require billing details be sent for a successful transaction. AVS is one example of this, but there could be more. This option depends entirely on the setup on Moneris.

These are the production credentials

The storeID and API token are supplied by Moneris

The checkoutID is found on the Moneris Checkout Config screen

Hosted Token ID is found on the Hosted Tokenization screen after creating a profile.

These are only used when testing on the preview server.

Before beginning the configuration, you must create a profile in your Moneris account. From there, you will need to obtain the following credentials from within the Moneris account. OPSCOM will require these details to connect to the service.

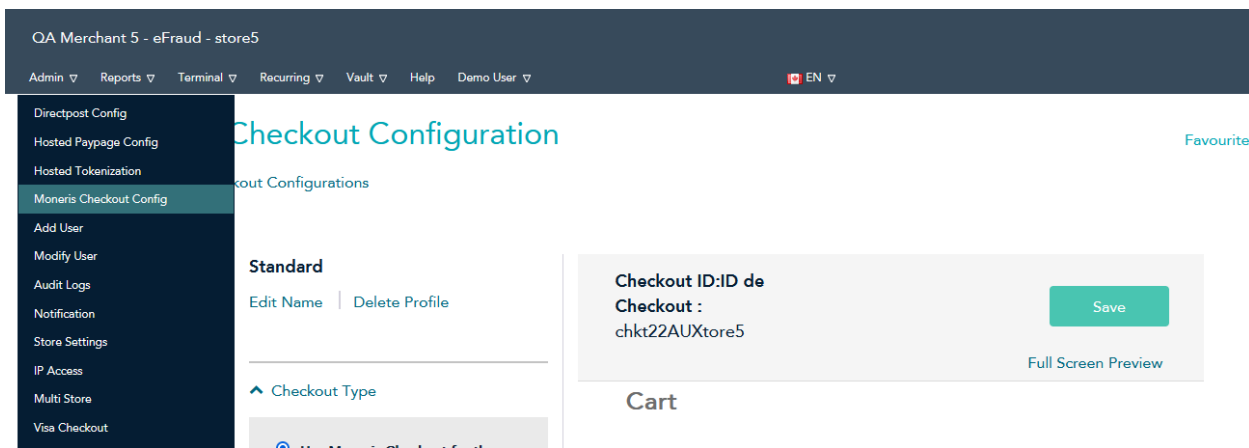
- store\_id: store#
- api\_token: xxXXXXXX
- checkout\_id: xxxxxxxxStore#
- Hosted\_token\_id (Generated by you, later on in this document)

<div class="info-box"> You must also determine if your Moneris security settings, such as Address Verification Service (AVS), will require your customer's billing address to be collected. This decision affects which fields OPSCOM will prompt the user to complete.

</div>

## Moneris Portal Configuration

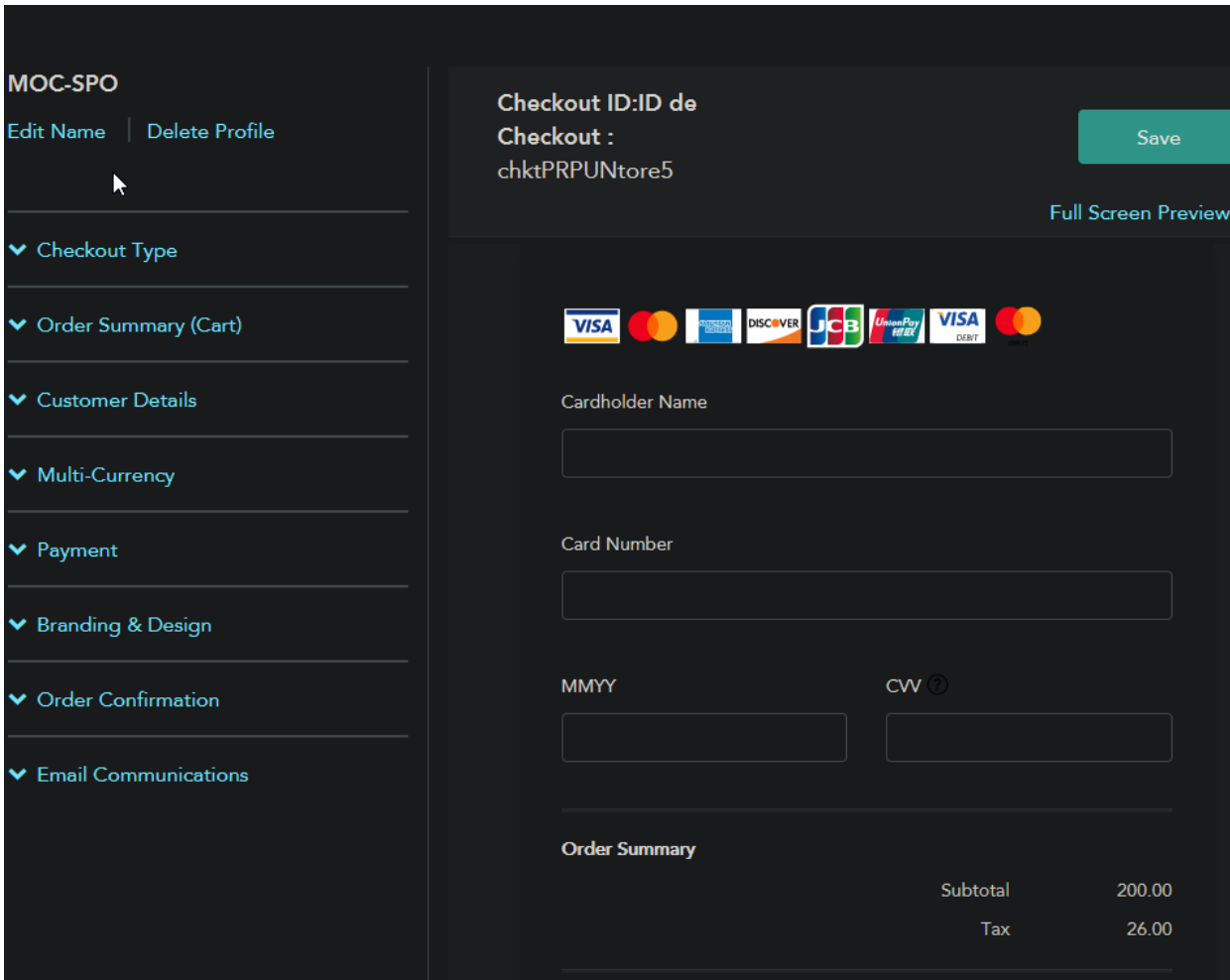
1. Log in to the Moneris Merchant Resource Center and navigate to the configuration area.



2. Click Admin in the main menu and select Moneris Checkout Config.
3. Locate your checkout profile (e.g., MOC-SPO) and click to edit it.



From there, you can change any of the settings:



#### Checkout Type

- Use Moneris Checkout for the complete set of online payment forms.

Select this option if you want to use Moneris forms for order summary, payment, shipping details, and order confirmation.

- I have my custom order form and want to use Moneris simply for payment processing.

Select this option if you want to embed Moneris payment details within your own order form.

## Checkout Type

### Select;

“I have my custom order form and want to use Moneris simply for payment processing.”

This ensures that Moneris provides the payment fields embedded within your existing OPSCOM order form.

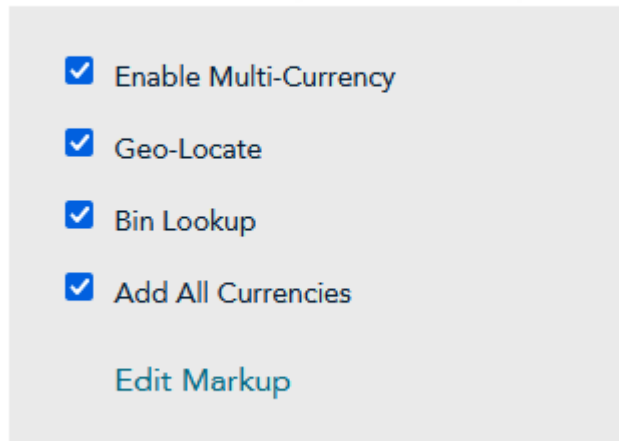
## Multi-Currency

---

This feature

### Multi-Currency

payments in multiple



Enable Multi-Currency

Geo-Locate

Bin Lookup

Add All Currencies

Edit Markup

currencies.

---

**Feature Limitations:** with Multi-Currency Enabling Multi-Currency will disable support for Recurring Billing, Gift Cards, 3-DSecure 1.0, and Google Pay.

## Payment

This section defines transaction rules and security options.

- **This is where you can control whether or not you wish to use a pre-authorize or payment capture model, the benefits of which are described above.**
- **Payment Security:** It is your decision whether to enable security features like AVS and CVV. **If you enable these, you must configure OPSCOM to collect the full billing address from users.** Note that if

Multi-Currency is active, 3D Secure is not supported.

- **Transaction Type:** You can set this to either preauthorize, or purchase. **THE PAYMENT TYPE IN OPSCOM MUST ALSO BE CONFIGURED THE SAME WAY.**
- **Tokenize Card:** Ensure the Tokenize Card option is ***disabled***. Card tokenization for recurring payments is handled by the **Moneris Vault**, which is configured separately.

The purchase model (current) requires a client's connection be stable. Should anything interrupt that, javascript errors, spotty connection, etc., the callback will not reach OPSCOM. With the new pre-auth model, the capturing of the funds now happens server to server meaning neither the client, the app, nor spotty connections can interfere with processing the payment. The Preauthorize Capture model provides a more reliable payment process.

## Branding and Design

To ensure the payment form integrates smoothly with the user experience:

- Disable the Enable Fullscreen checkbox. This prevents the checkout form from taking over the entire screen.

The rest of the branding and design decisions can be configured to your liking.

## Order Confirmation

Configure Moneris to redirect the user back to OPSCOM after a payment attempt.

- Select the option “Use Own Page” for the order confirmation. This allows OPSCOM to display the final success or failure receipt.
- 

## OPSCOM System Configuration

### Payment Types Setup

Please click this [hyperlink](#) to be redirected to our Payment Types Setup documentation, and then finish the remaining steps after that has been finished.

Create a new section for setting up Hosted providers. We currently have only three:

- [OpenEdge](#)
  - [Moneris Checkout](#)
  - [Bambora Checkout](#)
- 

### Moneris Vault for Stored Payments

The screenshot shows the Moneris Admin interface for 'QA Merchant 5 - eFraud - store5'. The 'Hosted Tokenization' menu is open, listing options such as Directpost Config, Hosted Paypage Config, Hosted Tokenization (highlighted), Moneris Checkout Config, Add User, Modify User, Audit Logs, Notification, Store Settings, IP Access, Multi Store, Visa Checkout, Apple Pay, and Google Pay.

Below the menu, the 'Existing Profiles' section contains a table with the following data:

	Profile Id	
Remove	ht32X7Z15CKFDIG	
Remove	ht3C25NKE934ZJB	
Remove	ht3CBEPXGU4BE6E	http://test.com

Moneris Vault is a complementary service used for tokenizing and storing credit cards for future use, such as with parking rollovers or Text2ParkMe. This functionality requires one additional piece of information during setup:

- `hosted_token_id`: This ID is generated from the Hosted Tokenization page within the Moneris portal.

If you encounter an "Invalid credentials" error when attempting to tokenize a card, it is likely due to a missing `hosted_token_id` or because the token's "Source URL" authentication does not match the domain you are using. For

testing, you can use a profile without Source URL authentication.

---

## Best Practices and Troubleshooting

### Common Billing and AVS/CVV Errors

The most common issue during setup is a mismatch between Moneris's expectations and the data OPSCOM provides.

If Moneris is configured to require and validate billing details (AVS is enabled), but OPSCOM is only configured to ask for an email address (promptFor is set to anonymous), all transactions will fail. You must ensure the OPSCOM configuration matches the security requirements you set in the Moneris portal.

---

Take Command of Your Parking and Security - <https://OperationsCommander.com>

Revision #17

Created 14 May 2024 08:05:14

Updated 17 March 2026 08:50:45