



# Moneris® Go.

## Integration Guide



Documentation Version: 1.4.0

Documentation Date: June 19, 2025

Copyright © Moneris Solutions, 2025

All rights reserved. No part of this publication may be reproduced, stored in retrieval systems, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Moneris Solutions Corporation.



## Table of contents.

1.	Document Control .....	3
2.	Introduction to Integration Solutions.....	5
2.1	Obtaining integration documentation.....	6
2.1.1	Other documentation resources available online.....	6
3.	Communication .....	7
3.1	Integration modes and communications supported.....	7
3.2	Preparing your firewall (Wi-Fi and Ethernet connection).....	7
4.	Getting Started .....	8
4.1	Development resources from Moneris.....	8
4.2	SYNC.....	9
4.3	BIN Ranges Whitelist ( <i>Requires Moneris Approval</i> ).....	9
5.	Implementing App to App Integration ( <i>Coming Soon</i> ).....	10
5.1	Summary of the process.....	11
5.2	Sync with Moneris.....	11
5.3	Develop and test your android application .....	12
5.3.1	Application requirements.....	12
5.4	Promote your application to production.....	12
6.	Implementing Cloud Integration .....	13
6.1	Cloud requirements .....	13
6.2	Hardware and cloud integration device setup.....	13
7.	Implementing Direct Integration.....	14
7.1	Direct integration pre-requisites.....	14
7.2	Hardware and direct integration device setup.....	14
8.	Tip Functionality .....	15
8.1	Tip Functionality Supported Devices.....	15
8.2	Tip on Subtotal .....	16
9.	Taxes and Flat Fees .....	17
9.1	Tax Functionality Supported Devices.....	17
10.	Tokenization on Moneris Go Terminals ( <i>Coming Soon</i> ) .....	19
10.1	Implementing Tokenization.....	19
10.2	Tokenization Supported Devices.....	19
10.3	Tokenization Flow.....	20
11	Dynamic Currency Conversion (DCC) on Moneris Go Terminals.....	22
11.1	Implementing Dynamic Currency Conversion (DCC) .....	22
11.2	Dynamic Currency Conversion (DCC) Supported Devices .....	22
11.3	Enabling Dynamic Currency Conversion (DCC) .....	22

11.4	Supported Cards and Interface .....	22
11.5	User experience .....	23
11.6	What Integrators need to know? .....	23
12.	Offline Payments on Moneris Go Terminals .....	26
12.1	Processing Offline Payments.....	26
12.2	Offline Payments Supported Devices.....	27
12.3	Offline Payments Configuration .....	27
12.4	What do Integrators need to know?.....	28
13.	Installments Enabled by Visa on Moneris Go Terminals .....	29
13.1	Supported Terminals .....	29
13.2	Enabling Installments by Visa .....	29
13.3	Supported Cards and Interface .....	29
13.4	User Experience .....	30
13.5	What Integrators need to know? .....	30
14.	Configure Terminal Settings .....	33
14.1	Configuration Scope .....	33
14.2	Configurable Settings via Sync Request.....	33
14.3	Validation Rules .....	33
15.	Support.....	34

# 1. Document Control

Version	Date	Section	Summary of changes
1.0	Aug-10-2023	All sections	Initial release
1.0.2	Oct-12-2023	5.0 Implementing App to App Integration	Whitelist of IPs and Track
1.1.2	Mar-6-2024	8.0 Implementing additional features on Go terminals	This is a new section
		2.1 Obtaining integration documentation	This is a new section
		3.1 Integration modes and communications supported	Updated chart
		5.0 Implementing App to App Integration	Updated Ingenico – App to App Integration section
1.2.2	Apr-30-2024	8.0 Implementing additional features on Go terminals	Renamed section to Tokenization on Moneris Go Terminals
		9.0 Dynamic Currency Conversion (DCC) on Moneris Go Terminals	This is a new section
		10.0 Store and Forward (SAF) on Moneris Go Terminals	This is a new section
		11.0 Installments Enabled by Visa on Moneris Go Terminals	This is a new section
1.3.2	Jul-23-2024	Title page	Updated device images to include Moneris Go PIN Pad
		All sections	Updated Unified Gateway/eCommerce API References
		4.1 Getting Started	Wording corrections
		4.2 SYNC	This is a new section
		4.3 BIN Ranges Whitelist	This is a new section
		5.0 Implementing App to App Integration	Moved content around for clarity
		10.0 Store and Forward (SAF) on Moneris Go Terminal	Renamed section to Offline Payments on Moneris Go Terminal and updated references
		10.3 Store and Forward Configuration	Wording corrections
1.3.3	Nov-29-2024	10.0 Offline Payments on Moneris Go Terminals	Updated reference from Store and Forward (SAF) to Offline Payments
		10.1 Implementing Tokenization and 10.3 Tokenization Prerequisites	Updated content for clarity
		10.4 Offline Payments - API Integration	This is a new section

1.3.4	Mar-24-2025	8. Enable Tip Functionality	This is a new section
		9. Enable Tax Functionality	This is a new section
1.4.0	Jun-19-2025	3.1 Integration modes and communications supported	Updated chart
		5.2 Contact Moneris	Renamed section to Sync with Moneris
		6.1 Cloud requirements	Added link for Moneris Unified Gateway API Sec doc
		8. Enable Tip Functionality	Renamed section to Tip Functionality and updated content
		9. Enable Tax Functionality	Renamed section to Taxes and Flat Fees
		10.2 Tokenization Supported Devices	Updated chart
		10.3 Tokenization Prerequisites	Renamed section to Tokenization Flow
		11.2 Dynamic Currency Conversion (DCC) Supported Devices	Updated chart
		11.6 What Integrators need to know?	Updated section for content clarity and updated Sample trace and receipt
		12.0 Offline Payments on Moneris Go Terminals	Updated sections for content clarity
		12.2 Offline Payments Supported Devices	Updated chart
		12.4 What do Integrators need to know?	Updated section for clarity
		13.1 Supported Terminals	Updated chart
		13.5 What Integrators need to know?	Updated section for clarity
		14.0 Configure Terminal Settings	This is a new section

## 2. Introduction to Integration Solutions

This guide will help you to integrate your secure, PCI-compliant Moneris Go terminal with your own application(s). This will allow your customers to respond to prompts on screen (including tip prompts), and to safely enter their card information. There are three available modes of integration.

### *App to App Integration*

In this mode, you build your own **Android application** for your customers to use on the Moneris Go terminal. When customers want to make an in-app purchase, your application connects to the Moneris payment application on the terminal using our application programming interface (API). The Moneris payment application takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit card(s).

To integrate Moneris Go with your own Android application, please refer to ***Moneris Go - App to App and Direct Integration - API Specification*** document.

### *Cloud Integration*

In this mode, you build your own **point of sale (POS) application** which runs on a separate device. Your POS application connects to the Moneris terminal through the internet and Moneris' Cloud server. You start transactions on your POS application. Transaction requests are sent through the internet to Moneris' Cloud server which routes the transaction to the appropriate payment terminal. The Moneris payment application on the terminal takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit card(s).

To integrate Moneris Go with your POS application, please refer to ***Moneris Go Cloud 3.0 - API Specifications Cloud API*** document.

### *Direct Integration*

In this mode, you build your own **point of sale (POS) application** which runs on a separate device. Your POS application connects to the Moneris terminal over your local network using Wi-Fi or Ethernet. Transactions are started on your POS application which passes the transaction request to the terminal. The Moneris payment application on the terminal takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit cards(s).

To integrate Moneris Go with your POS application, please refer to ***Moneris Go App to App and Direct Integration - API Specification*** document.

## 2.1 Obtaining integration documentation

The Moneris Client Consulting team provides merchants and integrators with the following documentation:

- Moneris Go - App to App and Direct Integration - API Specification v0.xx
- Moneris Go Cloud 3.0 - API Specification v3.xx
- Moneris Unified Gateway API Specification
- Moneris Unified eCommerce API Specification

### 2.1.1 Other documentation resources available online

Here are other support resources that are available online:

- Moneris has its own GitHub page, which contains APIs for Moneris eCommerce and Gift & Loyalty products; visit <https://github.com/orgs/Moneris/repositories?type=all>
- The Moneris Developer Portal contains information on how to integrate, a list of APIs and documentation, DTD or Java .NET & PHP libraries, and FAQs; visit <https://developer.moneris.com/>
- You can access PAX resources via the PAX developer portal; visit <https://developer.pax.us/>
- You can access Ingenico resources via the Ingenico developer portal; visit <https://ingenico.com/ca-en/developers>

## 3. Communication

Moneris Go terminals can communicate in various ways.

### 3.1 Integration modes and communications supported

Device Model	Mode>>>	Direct			App to App			Cloud		
		Ethernet	Wi-Fi	Cellular	Ethernet	Wi-Fi	Cellular	Ethernet	Wi-Fi	Cellular
Moneris Go - PAX A920		N/A	✓	N/A	N/A	✓	✓	N/A	✓	✓
Moneris Go Unattended - PAX IM30		✓	✓	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Moneris Go PIN Pad - PAX A35		✓	✓	N/A	N/A	N/A	N/A	✓	✓	N/A
Moneris Go Plus - DX8000		N/A	✓	N/A	N/A	✓	✓	N/A	✓	✓
Moneris Go Slim - EX8000		N/A	✓	N/A	N/A	✓	✓	N/A	✓	✓

1. Functional base required for Ethernet

The terminal can communicate with Moneris, your Electronic Cash Register (ECR) application or manufacturer provided services using any of the communications supported in the above table. When using ethernet or Wi-Fi communication, it would be your responsibility to configure the network so Moneris terminal(s) can communicate with required URL's.

### 3.2 Preparing your firewall (Wi-Fi and Ethernet connection)

Please refer to [Merchant Network Readiness Checklist](#)





## 4. Getting Started

You will need to ensure you have the necessary items for both test and production environments.

### 4.1 Development resources from Moneris

Check that you have the following items:

- Debug terminal (for App to App only)
- QA Terminal (for Cloud and Direct only)
- API specification for your desired integration mode
- A set of Moneris test cards
- Moneris account credentials for test and production:

Environment	Account credentials
<b>Test</b>	<ul style="list-style-type: none"> <li>▪ Merchant ID (MID) consisting of 13 numeric digits beginning with 0030</li> <li>▪ Associated Store ID (alphanumeric, typically beginning with mogo- or gwca-)</li> <li>▪ Terminal ID (TID) consisting of 8 characters</li> <li>▪ QA Go Portal account (<b><i>required prior to device setup</i></b>)</li> <li>▪ API token (for Cloud and to support Tokenization)</li> </ul>
<b>Production</b>	<ul style="list-style-type: none"> <li>▪ Merchant ID (MID) consisting of 13 numeric digits beginning with 0030</li> <li>▪ Associated Store ID (alphanumeric, typically beginning with mogo- or gwca-)</li> <li>▪ Production Terminal ID (TID) consisting of 8 characters</li> <li>▪ Production Go Portal account (<b><i>required prior to device setup</i></b>)</li> <li>▪ API token (for Cloud and to support Tokenization)</li> </ul>

If anything is missing, contact your Moneris Client Consultant for assistance with ordering equipment and setting up the related test account(s).

## 4.2 SYNC

It is very important that ECR and/or applications reside on Moneris devices that are integrated using App to App integration and must support SYNC. SYNC is required in order for the Moneris Go devices to obtain confirmation changes made on the host. These changes may include card plan changes, enabling/disabling various transaction types and/or features (for example: enable/disable TIP). It is recommended to support the “Default” command, but you may build other options in the ECR/APP if required.

## 4.3 BIN Ranges Whitelist (*Requires Moneris Approval*)

Work with Moneris’ teams for the following tasks:

**Whitelist Track<sup>1</sup>** (Note: This will be available for future release)

The API command `getTrackData` can be leveraged to read track2 data. In order to read and accept third party Gift and Loyalty card track2 data using the terminals card reader, you will be required to work with Moneris representative to whitelist the BIN range(s) for those cards. Moneris will require the low and high BIN range along with length of each BIN range. Example: 625173-625173(16) or 62162434626-62162434626(19).

<sup>1</sup>It is your responsibility to inform Moneris should there be any change to the bin-ranges. Moneris is not held responsible should there be any change. Moneris turn-around time to implement bin-ranges is approximately 2 to 4 weeks assuming all accurate data has been provided.

## 5. Implementing App to App Integration (*Coming Soon*)

You can incorporate App to App integration by following a few steps.

### **Whitelist of IPs<sup>1</sup>**

Moneris provided SIM cards are not exposed to all IP's on the internet. For any third party applications operating on Moneris terminal utilizing Moneris provided SIM's will require whitelist of IP addresses (associated to URLs) or endpoints. Communication to IP's not explicitly whitelisted by Moneris is blocked. Integrators are expected to test their connectivity with SIM cards from all supported Telcos during their integration phase. It is also your responsibility to ask for a SIM Card from each telco Moneris supports.

Duration to Whitelist IP's: it may take 6 to 8 weeks to whitelist IP's with the telco's. It is Integrator's responsibility to inform Moneris of all IP changes to avoid any impacts to your customers.

Moneris does not support IP's for individual stores/locations. IP address whitelisted must not exceed more than 10 IP's per integrator.

- Moneris also doesn't support URL based whitelisting. While we may ask for URL's associated to IP's, they are used for reference purposes.

### ***PAX – App to App Integration***

Using app to app integration, you can deploy your own Android application on Moneris terminals allowing your customers to make purchases quickly and securely. Follow this guide to:

- Get access to PAX resources and receive your debug terminal(s) – communication with the Client Consultant.
- It is the integrator's responsibility to provide list of IP's associated to URL's.
- Test your application within your PAX test store.
- Once your application is successfully deployed into the PAX Global marketplace, please work with Moneris representatives to subscribe to the application – adding it to the Moneris PAX test and production marketplace.

<sup>1</sup>Please work with your Moneris representative to whitelist all IPs your application(s) will be communicating with. This includes middleware and APIs. While our telecom partners only have the ability to whitelist by IPs, Moneris will still require the URL. It is your responsibility to inform Moneris should there be any change to IP addresses. Telcos require 8 to 10 weeks to implement change. These timelines may change due to holiday season or any telco's embargo in affect.

## ***Ingenico – App to App Integration***

Using app to app integration, you can deploy your own Android application on Moneris terminals allowing your customers to make purchases quickly and securely. Follow this guide to:

- Get access to Ingenico resources and receive your debug terminal(s) – communication with the Client Consultant.
- Provide your signed test application to Moneris. Moneris will upload the app to the QA Ingenico store. You can then download the application on your QA devices for testing.
- Once tested, please work with Moneris representatives to promote the application to the production store.

<sup>1</sup>Please work with your Moneris representative to whitelist all IPs your application(s) will be communicating with. This includes middleware and APIs. While our telecom partners only have the ability to whitelist by IPs, Moneris will still require the URL. It is your responsibility to inform Moneris should there be any change to IP addresses. Telcos require 8 to 10 weeks to implement change. These timelines may change due to holiday season or any telco's embargo in affect.

## **5.1 Summary of the process**

Below are the high-level steps in the process:

- **Development and testing:**
  1. Develop your own application on a test (debug) terminal from Moneris (refer to section [4.1 Development resources from Moneris](#) for more information).
  2. Upload test application to the PAX test marketplace. Refer to the [PAX developer centre](#) for instructions.
- **Production:**
  1. Upload your production application to the PAX production marketplace.
  2. PAX signs your application and makes it available to customers.
  3. Moneris subscribes to your application so that it can be downloaded to Moneris terminals.
  4. Download your application from the Moneris QA and Production PAX marketplace to your production Moneris terminal.

Review each section below to understand how to proceed.

## **5.2 Sync with Moneris**

Once you contact [Moneris](#), you will be assigned a Client Consultant who will coordinate your development journey. Moneris will set up a call with PAX to ensure you receive:

- The PAX non-disclosure agreement
- A PAX legal agreement for each development (debug) terminal you request
- Access to the PAX developer portal and test marketplace

Once you sign and return all legal agreements to PAX, Moneris will send you the debug terminal(s) and test cards.

## 5.3 Develop and test your android application

In addition to the standard Android development tools, you will need the Moneris Go API specification to call the Moneris Go payment application. Please refer to your Moneris Client Consultant on how to obtain the Moneris App to App developer documentation, a PAX A920 debug terminal and test cards.

### 5.3.1 Application requirements

Your application must conform to the following requirements:

- Printing done using Moneris' receipt API. Receipts must be stored electronically so they can be reprinted if required
- Payment transaction data must be retained for 24 months

**Note:** *This list is not exhaustive. Refer to the Moneris API specification for complete requirements.*

## 5.4 Promote your application to production

Follow the steps below to make your application available in the terminal's app store.

### **PAX**

In order for the application to be made available in the PAX store, you are required to upload the application to PAX's production PAX store marketplace. Once PAX has signed the application, it would be your responsibility to inform your Moneris representative. Your Moneris representative will then inform the appropriate group to execute the application subscription process<sup>1</sup>.

Once Moneris has subscribed to the application, it would be made available for download on all Moneris devices<sup>2</sup>.

<sup>1</sup>This task may take 1 to 2 weeks

<sup>2</sup>Moneris doesn't have the capability to push updates to sets of terminals. It would be your responsibility to inform the merchants so they can update the application.

### **Ingenico**

In order for the application to be made available in the Ingenico app store, you are required to sign the application with the tools provided by Ingenico (If the integrator does not have an Ingenico signing kit, obtain signing card from Ingenico<sup>1</sup>).

Once the application has been signed, you will work with Moneris representative to get the application loaded to the Ingenico app store<sup>2</sup>.

<sup>1</sup>This task may take 6 to 8 weeks

<sup>2</sup>Moneris doesn't have the capability to push updates to set of terminals. It would be your responsibility to inform the merchants so they can update the application.

## 6. Implementing Cloud Integration

This section describes how to put your Moneris Go terminal into cloud integration mode. Once the terminal connects to the Moneris cloud, you can send transactions from your point of sale (POS) software to the terminal for processing. Your customers can use the terminal to securely enter their payment cards and leave tips.

### 6.1 Cloud requirements

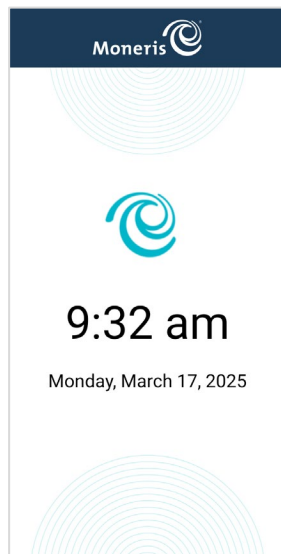
- Gateway store ID and API token
  - For **QA**, refer to your Client Consultant to obtain your store ID and API token
  - For **Production**, your store ID and API token are available through Merchant Direct

***Note:** QA and Production credentials will differ.*

- Moneris Go Cloud 3.0 - API Specification v3.xx
- [Moneris Unified Gateway API Specification documentation](#)

### 6.2 Hardware and cloud integration device setup

Please refer to your device's specific getting started guide for instructions on how to setup your payment terminal in Cloud operating mode. Once initial setup has been complete, simply power on your device to sync the settings with Moneris.



When you see the idle screen (showing the Moneris logo with the date and time), the device is ready to process transactions sent from your POS software.

## 7. Implementing Direct Integration

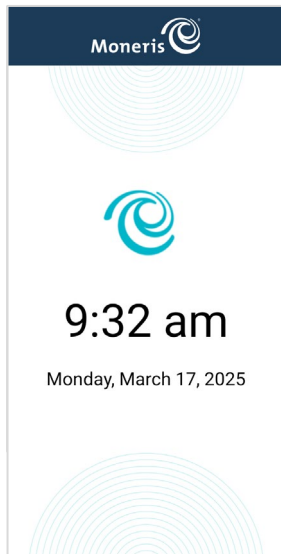
This section describes how to put your Moneris Go terminal into direct integration mode. You can send transactions from your point of sale (POS) software to the terminal for processing over your local area network. Your customers can use the terminal to securely enter their payment cards and leave tips.

### 7.1 Direct integration pre-requisites

- Local IP address and port of your terminal for communication
- Moneris API documentation (Moneris Go App to App and Direct documentation, *optional* Moneris Unified Gateway API Specification)
- Usage of the optional Unified Gateway API will require your Gateway store ID and API token

### 7.2 Hardware and direct integration device setup

Please refer to your device's specific getting started guide for instructions on how to setup your payment terminal in Direct operating mode. Once initial setup has been complete, simply power on your device to sync the settings with Moneris.



When you see the idle screen (showing the Moneris logo with the date and time), the device is ready to process transactions sent from your POS software.

## 8. Tip Functionality

Moneris Go terminals can now be set up to allow cardholders to tip based on either the subtotal (pre-tax amount) or the final total, which includes taxes.

To enable this feature, please contact your Moneris Client Consultant.

### 8.1 Tip Functionality Supported Devices

Product Name - Device Model	Operational Modes		
	Direct	App to App	Cloud
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A
Moneris Go PIN Pad - PAX A35	✓	✓	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓




## 8.2 Tip on Subtotal

Merchants have the option to tip based on the pre-tax (subtotal) amount of their bill, or the final total that includes taxes. Contact your Moneris Client Consultant to enable this feature.

The ECR application must include the subtotal in the purchase request message so that the terminal can calculate the tip based on the pre-tax amount.

Example of **Tip on Purchase total** (with taxes and fees) and **Tip on Subtotal** (without taxes and fees):



Would you like to leave a tip?

Purchase total

**\$ 11.50**

10%  
\$ 1.15


15%  
\$ 1.73

20%  
\$ 2.30

\$

%

No Tip



Would you like to leave a tip on the subtotal?

**\$ 10.00**

Total: \$ 11.50

10%  
\$ 1.00

15%  
\$ 1.50

20%  
\$ 2.00

\$

%

No Tip

## 9. Taxes and Flat Fees

All Moneris Go devices can print taxes and flat fees in receipts. You can send up to 5 taxes and fees for each financial transaction through the taxes array in the API. Refer to Section 7.2 to 7.8 in the App to App and Direct Integration as well as Cloud Integration API Specification.

### 9.1 Tax Functionality Supported Devices

Product Name - Device Model	Operational Modes		
	Direct	App to App	Cloud
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A
Moneris Go PIN Pad - PAX A35	✓	✓	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓

Sample purchase request with various taxes and fees along with the equivalent receipt:

```

"data": {
  "request": [
    {
      "idempotencyKey": "example_idempotencyKey",
      "action": "purchase",
      "totalAmount": "3000",
      "subtotalAmount": "2000",
      "taxes": [
        {
          "taxName": "GST",
          "taxAmount": "100"
        },
        {
          "taxName": "HST",
          "taxAmount": "200"
        },
        {
          "taxName": "PST",
          "taxAmount": "300"
        },
        {
          "taxName": "QST",
          "taxAmount": "300"
        },
        {
          "taxName": "Eco Fee",
          "taxAmount": "100"
        }
      ]
    }
  ]
}

```

---- DOSSIER DE TRANSACTION ----	
MAGASIN DE BONBONS DEE 3300, RUE BLOOR OUEST ETOBICOKE ON	
Achat	
20 fév 2025	12:26:04
INTERAC	*****3412
CHÈQUE	
Saisie : Puce (C)	
#Ref : 488-0STA0BG3S3ANWLW	
# d'autor : 420952	Réponse : 00-001
ID de commande	MG01742228763307
Sous-total	20,00 \$
GST	1,00 \$
HST	2,00 \$
PST	3,00 \$
QST	3,00 \$
Éco-frais	1,00 \$
Montant	30,00 \$
Pourboire	4,00 \$
Total	34,00 \$
A0000002771010 Interac TVR 805000000 TSI 6800	
Approuvé VÉRIFIÉ PAR NIP	

The **totalAmount** and **subtotalAmount** fields must be provided in cents in the API request. For example, a total amount of \$30.00 should be sent as "**totalAmount**": "3000", and a subtotal amount of \$20.00 should be sent as "**subtotalAmount**": "2000".

The **taxName** parameter is dynamic, meaning the integrator can pass any applicable tax name relevant to their business. Examples include "HST" or "Harmonized Tax".

The **taxAmount** must be specified as an absolute value, meaning the dollar amount must be converted to cents. It should not be passed as a percentage.

Please note that we do not calculate or validate tax values. The amounts provided in the transaction request will be printed on the receipts in dollar value.

## 10. Tokenization on Moneris Go Terminals *(Coming Soon)*

### 10.1 Implementing Tokenization

This section describes how to implement tokenization features on your Moneris Go terminal for Direct, App to App and Cloud integration mode. Once the integration has been completed and the tokenization feature is enabled, you need to leverage the Unified eCommerce API in order to create, store and manage your tokens via the Moneris Vault.

### 10.2 Tokenization Supported Devices

Product Name - Device Model	Operational Modes		
	Direct	App to App	Cloud
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	✓	N/A	N/A
Moneris Go PIN Pad - PAX A35	✓	N/A	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓

#### Tokenization Prerequisites

- Gateway store ID and API token
  - For **QA**, refer to your Client Consultant for assistance enabling Tokenization on your existing QA store if not enabled or for assistance obtaining a QA store if one does not exist
  - For **Production**, your store ID and API token are available through Merchant Direct

**Note:** QA and Production credentials will differ.

- Moneris Go API documentation (refer to [section 2.1.1](#))
  - Moneris Go Cloud 3.0 - API specification
  - Moneris Go App to App and Direct Integration - API specification
- Moneris Unified eCommerce API (refer to [section 2.1.1](#))
- DTD or JAVA, .NET, PHP libraries

## 10.3 Tokenization Flow

On your Moneris Go device, perform a Financial Transaction such as a Card Verification, Purchase or Pre-auth and retain the following information from a successful response:

- orderId,
- transactionId, alternatively known as transaction number

Financial Transactions include the following:

- Purchase
- Refund
- Void (aka Purchase Correction)
- Independent refund
- Pre-Auth
- Completion
- Card verification

Refer to **Moneris Go API** specifications for additional information.

Leveraging the Unified eCommerce API, perform a Vault Tokenize Credit Transaction request with the following object values:

- **Mandatory Object Values:**
  - orderId,
  - transactionId, alternatively known as transaction number

### Sample Code (Tokenize Credit Transaction):

```

1. <?xml version="1.0"?>
2. <request>
3.   <store_id>gateway storeID</store_id>
4.   <api_token>API token</api_token>
5.   <res_tokenize_cc>
6.     <order_id>order ID from Go transaction</order_id>
7.     <txn_number>transaction number from Go transaction (ex.1017-0_140)</txn_number>
8.   </res_tokenize_cc>
9. </request>

```

- **Optional Object Values:**
  - Customer ID
  - Data Key
  - Email Address
  - Phone Number
  - Note
  - AVS Information\*
  - COF Information

*\*When passing AVS information the following value are mandatory:*

- AVS street number
- AVE street name
- AVS zip/postal code

**Note 1:** A token cannot be generated in the event that a Moneris Go Device has been enabled for Offline Payments and a transaction was approved offline.

Refer to **Unified eCommerce API** for additional information associated with adding a token.

Leveraging the Unified eCommerce API for future financial transaction with credential on file:

- payment\_indicator: U
- payment\_information: 2

### Sample Code (Purchase with Token):

```

1. <?xml version="1.0"?>
2. <request>
3.   <store_id>gateway storeID</store_id>
4.   <api_token>API token</api_token>
5.   <status_check>>false</status_check>
6.   <res_purchase_cc>
7.     <data_key>Vault token</data_key>
8.     <order_id>example unique order ID</order_id>
9.     <amount>10.00</amount>
10.    <crypt_type>7</crypt_type>
11.    <cof_info>
12.      <payment_indicator>U</payment_indicator>
13.      <payment_information>2</payment_information>
14.    </cof_info>
15.  </res_purchase_cc>
16. </request>

```

**Note 2:** In the event of a pre-authorization via a Moneris Go Terminal followed by a token request, you have the option to leverage the token to perform a force post transaction, however, the orderID and transactionID must be sourced back to the original pre-auth transaction. Refer to the **Unified eCommerce API** or additional information associated with performing financial transactions via a token.

### Managing tokens:

You may leverage the Unified eCommerce API in order to perform the following token management functions:

- Vault Update Credit Card
- Vault Delete
- Vault Look Up Masked
- Vault Get Expiring
- Vault Add Token

Refer to the **Unified eCommerce API** for additional information associated with performing token management functions.

## 11 Dynamic Currency Conversion (DCC) on Moneris Go Terminals

### 11.1 Implementing Dynamic Currency Conversion (DCC)

This section describes how Dynamic Currency Conversion (DCC) functions on Moneris Go devices and how integrators can support it. DCC is a feature that allows cardholders the option to pay either in their native currency or in Canadian dollars.

### 11.2 Dynamic Currency Conversion (DCC) Supported Devices

Device Model	Dynamic Currency Conversion (DCC)		
Modes>>>	App to App	Cloud	Direct
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	✓
Moneris Go PIN Pad - PAX A35	N/A	✓	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓

### 11.3 Enabling Dynamic Currency Conversion (DCC)

Your client consultant can enable DCC on your Go Devices. You should ensure your device is on the latest software version and sync your terminal with Moneris in order to start using DCC. To reinitialize your terminal, refer to the Using Your Terminal section of the Moneris Reference Guide available at [moneris.com/en/support](https://moneris.com/en/support).

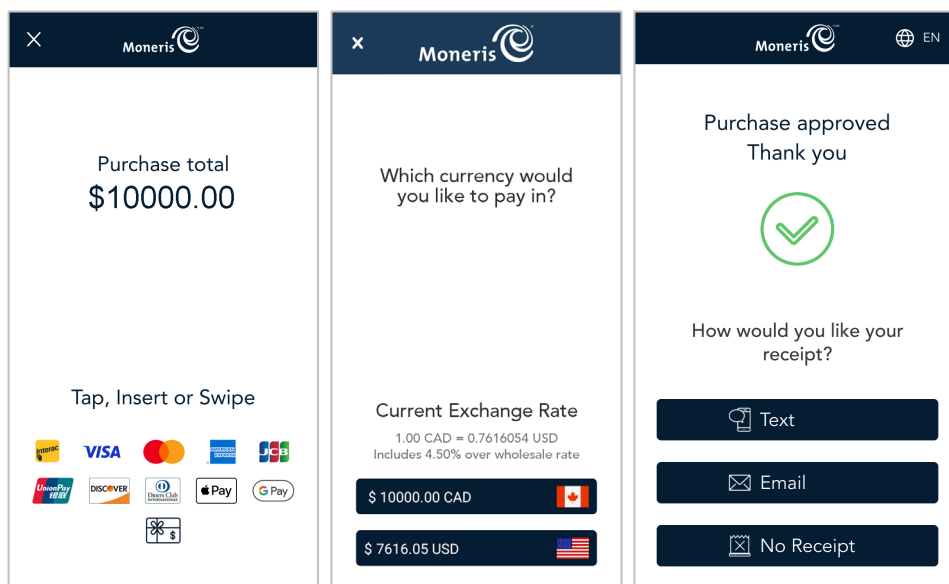
### 11.4 Supported Cards and Interface

DCC is available for Visa and Mastercard cards issued in a foreign currency for both Contact and Contactless interfaces.

**Note:** For Mastercard Contactless transactions, DCC is only offered for mobile wallets.

## 11.5 User experience

### Example of user screens:



## 11.6 What Integrators need to know?

- No development is required to support Dynamic Currency Conversion (DCC), and no specific parameters need to be sent by the Electronic Cash Register (ECR) to enable DCC on the device.
- An optional DCC opt-in parameter is returned in the response from the terminal payment application. This parameter is purely informational and indicates that the cardholder has opted into the DCC offer.
- DCC is available for purchase transactions, independent refunds, pre-authorization and completion transaction types. Note that DCC is not available for refund transactions.



### Sample trace:

Request:

```
{
  "apiVersion": "3.0",
  "apiToken": "BiZvfQiQl7Lohc6wem67",
  "storeId": "mogo000122",
  "polling": "true",
  "istConfigCode": "C123456SI",
  "dataId": "1742697337742-001",
  "dataTimestamp": "2025-03-22 22:35:37",
  "data": {
    "request": [
      {
        "idempotencyKey": "704283c7-a88c-4df3-9af6-867f528af7ca",
        "terminalId": "A4000007",
        "action": "purchase",
        "totalAmount": "100000"
      }
    ]
  }
}
```

**Response:**

```

"receipt": {
  "apiVersion": "3.0",
  "dataId": "1742697337742-001",
  "statusCode": "5207",
  "status": "Approved",
  "dataTimestamp": "2025-03-22 22:35:48",
  "data": {
    "response": [
      {
        "statusCode": "5207",
        "status": "Approved",
        "approvedAmount": "100000",
        "totalAmount": "100000",
        "cardType": "M ",
        "cardName": "MASTERCARD",
        "sequenceNum": "016",
        "realTimeUniqueId": "0STK1VOB9U0BNKS",
        "responseCode": "027",
        "iso": "01",
        "authCode": "T44766",
        "maskedPan": "*****0998",
        "orderId": "MG01742697342010",
        "transactionId": "784-0_1082",
        "idempotencyKey": "704283c7-a88c-4df3-9af6-867f528af7ca",
        "action": "purchase",
        "terminalId": "A4000007",
        "saf": "false",
        "dccOptIn": "true",
        "tenderType": "Credit",
        "formFactor": "14",
        "receiptChoice": "PRINT",
        "receipt": "
----- TRANSACTION RECORD ----- \r\n
GO CANDY STORE \r\n
3300 BLOOR ST W \r\n
ETOBICOKE ON \r\n
M8X2X2 \r\n
\r\n
Purchase \r\n
Mar 22,2025 22:35:48\r\n
MASTERCARD *****0998\r\n
\r\n
\r\n
Entry: Tap EMV (H) \r\n
\r\n
Ref#: 016-OSTK1VOB9U0BNKS \r\n
\r\n
Auth#: T44766 Response: 01-027\r\n
\r\n
Order: MG01742697342010\r\n
\r\n
\r\n
Amount $ 1 000.00\r\n
\r\n
\r\n
RATE* 1 CAD = 0.6763679 EUR \r\n
\r\n
*BASE RATE + 4.5% \r\n
\r\n
SELECTED TRANS CURRENCY \r\n
\r\n
[ ] CAD 1 000.00 CAD\r\n
\r\n
[X] EUR 676.37 EUR\r\n
\r\n
\r\n
Total 676.37 EUR\r\n
\r\n
\r\n
\r\n
\r\n
\r\n
A0000000041010 MCD05 v1 0 \r\n
\r\n
TVR 8000008001 \r\n
"completed": "true"
      }
    ]
  },
  "TxnName": "Purchase",
  "CloudTicket": "33ee6170-70d2-4760-8b67-3a0df4371c4f",
  "Completed": "true",
  "Error": "true"
}

```

## Sample DCC receipt:

```

----- TRANSACTION RECORD -----
DEE'S CANDY STORE
3300 BLOOR ST W
ETOBICOKE ON
M8X2X2

Purchase
Apr 01,2025 22:35:48
MASTERCARD *****0998

Entry: Tap EMV (H)
Ref#: 016-0STK1VOB9U0BNKS
Auth#: T44766 Response: 01-027
Order: MG01742697342010

Amount $ 1 000.00

RATE* 1 CAD = 0.6763679 EUR
*BASE RATE + 4.5%
SELECTED TRANS CURRENCY
[ ] CAD 1 000.00 CAD
[X] EUR 676.37 EUR

Total 676.37 EUR

A0000000041010 MCD05 v1 0
TVR 8000008001

Approved
Signature Not Required

Important:Retain this copy for your record

```

## 12. Offline Payments on Moneris Go Terminals

### 12.1 Processing Offline Payments

Offline payment functionality allows credit transactions to be completed without an active internet connection. This feature is useful when online payment processing is unavailable due to connectivity issues or system outages.

The terminal will encrypt and securely store the eligible financial transactions should the Moneris host become unreachable and authorize the transaction with the issuer. Once a connection is re-established, the terminal automatically uploads the transaction data to the host.

The following card bands are accepted offline:

- Visa
- Mastercard
- American Express
- JCB
- Discover

This functionality only supports Purchase, Independent Refund and Void\* transactions via the Chip insert interface.

#### Key points:

- \*Offline Purchase Void transactions will only Void Purchase transactions that were authorized offline.
- Refunds and Void transactions cannot be processed when the terminal is offline. An alternative way to refund the cardholder's funds is to trigger an independent refund transaction.
- Before authorizing an offline transaction, the payment application performs a risk assessment. Some of the checks that are performed are that the card is not expired, the cardholder entered the correct PIN, the card is a domestic card, the transaction doesn't exceed the offline transaction limit, and a series of other checks.

## 12.2 Offline Payments Supported Devices

Device Model	Offline Payments		
Modes>>>	App to App	Cloud	Direct
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	✓
Moneris Go PIN Pad - PAX A35	N/A	✓	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓

## 12.3 Offline Payments Configuration

The settings below must be configured on the Moneris Terminal Management System to enable offline payments functionality on your terminal.

For assistance, please contact your Moneris Service Manager or Client Consultant.

**Enable Offline Payments:** This flag indicates the ability to enable offline functionality. The default is disabled.

**Transaction Limit:** This limit indicates the maximum purchase transaction authorized value permitted in offline payments mode. The default limit is \$250.00. For example, if the maximum transaction amount is \$100.00, transactions exceeding this amount are not approved in Offline Payments mode. This limit does not apply to Independent Refund Transactions.

**Maximum Record Count:** This indicates the maximum number of offline transactions allowed to be stored offline, including purchase and independent refund transactions, that can simultaneously be stored on a terminal. The default limit is 500.

**Release Delay:** This setting determines how long the terminal stays in offline mode before it attempts to send a transaction to the Moneris Host for online authorization. While the device is in offline mode, it prompts the user to insert their card and checks if the transaction can be authorized offline. Once the timer expires, the terminal will also attempt to upload any stored offline transactions to the host. The default release delay is five minutes. If the Release Delay value is zero, the terminal will always try to connect online before accepting any eligible offline transactions.

**Burst Relay:** This option adds a delay between offline transactions, which is automatically released to the host. The default burst delay is two seconds.

## 12.4 What do Integrators need to know?

The Electronic Cash Register (ECR) must send the 'safAllowed' parameter as true in the financial request message. This will allow the terminal payment application to determine whether to approve the transaction offline when the request cannot reach the Moneris host.

The financial response message contains two indicators: 'saf' and 'safCounter.' These indicators inform the ECR application if a transaction was approved offline and indicate the total number of offline transactions stored in the device's memory. This information is provided for informational purposes only.

When the Electronic Cash Register (ECR) sends an order ID in the financial request message, the terminal payment application will modify and add extra data to the order ID once a transaction is approved offline. It's important to take this into account when integrating with the Moneris Go Terminal Solution.

## 13. Installments Enabled by Visa on Moneris Go Terminals

Installments Enabled by Visa offers the cardholder (with the eligible card) the flexibility to pay for merchandise or services in equal payments over a defined period and provides the cardholder with the option: Buy Now Pay Later (BNPL) during purchase.

### 13.1 Supported Terminals

Product Name - Device Model	Operational Modes		
	Direct	App to App	Cloud
Moneris Go - PAX A920	✓	✓	✓
Moneris Go Unattended - PAX IM30	✓	N/A	N/A
Moneris Go PIN Pad - PAX A35	✓	N/A	✓
Moneris Go Plus - DX8000	✓	✓	✓
Moneris Go Slim - EX8000	✓	✓	✓

### 13.2 Enabling Installments by Visa

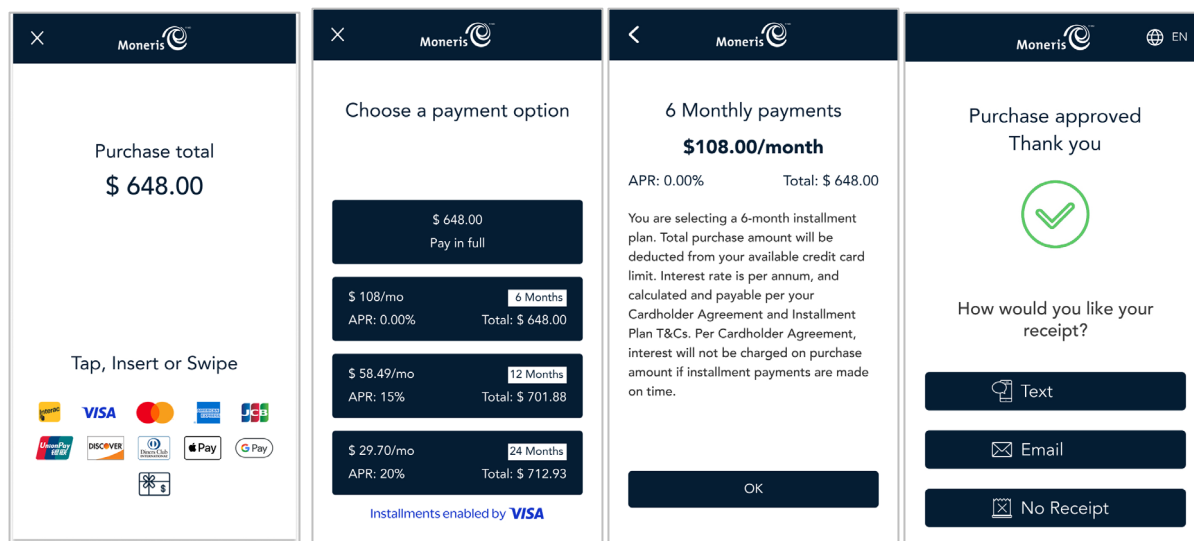
Your Client Consultant can help enable Installments by Visa on your Go Device. You should ensure your device is on the latest software version and sync your terminal with Moneris in order to start using Installments Enabled by Visa. To sync your terminal, refer to the Using Your Terminal section of the Moneris Reference Guide available at [moneris.com/en/support](https://moneris.com/en/support).

### 13.3 Supported Cards and Interface

This feature is available for Visa, Mastercard and American Express cardholders when inserted or tapped.

## 13.4 User Experience

Example of user screens:



## 13.5 What Integrators need to know?

- Installment plans are offered for Purchase and Pre-Authorization transactions. There are no specific parameters that need to be sent by the Electronic Cash Register (ECR) to enable Installments by Visa on the device.
- When a cardholder selects an installment plan, the terminal payment application returns an opt-in parameter in response to the ECR to indicate this selection.
- For Independent Refund transactions, it is IMPORTANT to send a real-time unique identifier parameter. This identifier must match the value received from the Purchase response.

## Sample Trace:

### Request

```
{
  "apiVersion": "3.0",
  "dataId": "1714074129526-001",
  "dataTimestamp": "2024-04-25 15:42:09",
  "data": {
    "request": [
      {
        "idempotencyKey": "51283cb9-f551-428a-8ce8-417c40877ffa",
        "terminalId": "A2345678",
        "action": "purchase",
        "totalAmount": "100000"
      }
    ]
  }
}
```

### Response

```
{
  "receipt": {
    {
      "apiVersion": "3.0",
      "dataId": "1714074129526-001",
      "statusCode": "5207",
      "status": "Approved",
      "dataTimestamp": "2024-04-25 15:42:29",
      "data": {
        "response": [
          {
            "statusCode": "5207",
            "status": "Approved",
            "approvedAmount": "100000",
            "totalAmount": "100000",
            "cardType": "V ",
            "cardName": "VISA",
            "sequenceNum": "010",
            "realTimeUniqueId": "0SCIK2O4R8MMDG0",
            "responseCode": "027",
            "iso": "01",
            "authCode": "B44893",
            "maskedPan": "*****3925",
            "orderId": "MGO1714074135255",
            "transactionId": "191-0_755",
            "idempotencyKey": "51283cb9-f551-428a-8ce8-417c40877ffa",
            "action": "purchase",
            "terminalId": "A2345678",
            "saf": "false",
            "installmentOptIn": "true",
            "tenderType": "Credit",
            "receiptChoice": "PRINT",
            "receipt": "      ----- TRANSACTION RECORD -----      \r\n                        COFFEE ..."
          }
        ]
      }
    }
  },
}
```



## Sample Installment by Visa receipt:

```

----- TRANSACTION RECORD -----

                                Purchase
May 08,2024                                07:29:54
MASTERCARD                                *****8118
UID: 0S3T1DU7UFUX4IK                    Entry: Chip (C)
Sequence: 005                                Batch: 016
Auth#: KN7980                                Response: 01-027

Amount                                $1,200.00
Monthly Payment                        $200.00
Number of Installments                  6
APR                                    12.99%
Total Cost                            $1,254.24
Total                                $1,200.00
A0000000041010
MASTERCARD
TVR 8000008000 TSI 6800
      Approved - Thank You
      Installments enabled by VISA

You are selecting a 6-month installment plan.® The total purchase amount will be deducted from your available credit limit. As set forth in your terms, your installment fee will be APR calculated for 6 months. If you miss an installment payment, the standard rate of purchases will apply to the remaining installment balance.

https://www.cibc.com/content/dam/personal\_banking/credit\_cards/PDF/cibc-pace-it-installment-plans-during-checkout-terms-and-conditions-en.pdf

Important: Retain this copy for your record

```

## 14. Configure Terminal Settings

Terminal settings can be configured by the ECR/POS using the Sync API. The Sync API enables the ECR to push configuration settings to the Terminal Management System (TMS), which then applies them to the terminals. Please refer to Section 5.3 of the Moneris Go - App to App and Direct Integration - API Specification v0.42 and Section 4.3 of the Moneris Go Cloud 3.0 - API Specification v3.18.

### 14.1 Configuration Scope

Settings can be applied at either the **Terminal level** or the **Merchant level**, but not both simultaneously. Mixing both levels will result in inconsistent behavior.

### 14.2 Configurable Settings via Sync Request

The following settings can be configured by the ECR/POS using the Sync Request:

- Tip
- Cashback
- Surcharge
- MOTO (Mail Order / Telephone Order)
- Custom field names
- Receipt functionality
- Restaurant settings (for Pay at the Table application only)
- Default terminal language

### 14.3 Validation Rules

1. All values received from the ECR must be in string format, including Boolean values ("true" or "false").
2. Any setting outside the supported list above will result in an error response.

## 15. Support

To obtain support for the terminal solution, follow these directions.

Situation	Directions
Your integration or development provider requires support related to the API specifications and tools.	Contact your Moneris Client Consultant.
You are facing difficulties uploading your application to the PAX marketplace	Contact your PAX representative.
You are facing difficulties uploading your application to the Ingenico marketplace	Contact your Ingenico representative.

NOTICE OF CONFIDENTIALITY. This document contains information that is the proprietary and confidential property of Moneris. The recipient agrees to maintain this information in confidence and not reproduce or otherwise disclose this information.

MONERIS, MONERIS BE PAYMENT READY & Design and MERCHANT DIRECT are registered trade-marks of Moneris Solutions Corporation. All other marks or registered trade-marks are the property of their respective owners.

© 2025 Moneris Solutions Corporation, 3300 Bloor Street West, Toronto, Ontario, M8X 2X2. All Rights Reserved. This document shall not wholly or in part, in any form or by any means, electronic, mechanical, including photocopying, be reproduced or transmitted without the authorized consent of Moneris Solutions Corporation. This document is for informational purposes only. Neither Moneris Solutions Corporation nor any of its affiliates shall be liable for any direct, indirect, incidental, consequential or punitive damages arising out of use of any of the information contained in this document. Neither Moneris Solutions Corporation or any of its affiliates nor any of our or their respective licensors, licensees, service providers or suppliers warrant or make any representation regarding the use or the results of the use of the information, content and materials contained in this document in terms of their correctness, accuracy, reliability or otherwise. Your credit and/or debit card processing is governed by the terms and conditions of your agreement(s) for merchant credit/debit card processing services with Moneris Solutions Corporation. It is your responsibility to ensure that proper card processing procedures are followed at all times. Please refer to the ***Moneris Merchant Operating Manual*** (available at: [moneris.com/support](https://moneris.com/support)) and the terms and conditions of your applicable agreement(s) for credit/debit processing or other services with Moneris Solutions Corporation for details.

MGo Int EN June 2025