Moneris SOLUTIONS

Merchant Integration Guide - Hosted Payment Page

Canada

Hosted Payment Page

INTERAC® Online Payment

Convenience Fee

ThreatMetrix

Hosted Tokenization

Hosted Vault

Version 1.0.0 - July 2016

Please Read Carefully

1. You have a responsibility to protect cardholder and merchant related confidential account information. Under no circumstances should ANY confidential information be sent via email while attempting to diagnose integration or production issues. When sending sample files or code for analysis by Moneris staff, all references to valid card numbers, merchant accounts and transaction tokens should be removed and or obscured. Under no circumstances should live cardholder accounts be used in the test environment.

2. The Transaction Risk Management Tool provides additional information to assist in identifying fraudulent transactions. In order to maximize the benefits from the Transaction Risk Management Tool it is highly recommended that you:
   a. Carefully consider the business logic and processes that you need to implement surrounding handling the response information the Transaction Risk Management Tool provides.
   b. Also implement the other fraud tools available through Moneris Gateway (e.g., AVS, CVD, Verified by Visa and MasterCard SecureCode).

3. When testing the Transaction Risk Management Tool there is specific test data that you will need to use. Please carefully review and follow the testing instructions and data provided in the document.

How Do I Get Help?

If you require technical assistance while integrating your store, please contact the Moneris Gateway Support Team:

For technical support:

Phone: 1-866-319-7450 (Technical Difficulties)

Email: onlinepayments@moneris.com

NOTE: To ensure that your call is directed appropriately please ensure that you have the 13 digit merchant number, (starts with 00302) associated with your account, available prior to calling and that you enter it correctly when prompted. This will allow us to direct your call to the specialized support team.

For integration support (8 am – 8 pm ET):

Phone: 1-866-562-4354

Email: eproducts@moneris.com

When sending an email support request please be sure to include your name and phone number, a clear description of the problem as well as the integration method you are using (e.g., Hosted Payment Page). For security reasons, please do not send us your ps_store_id combined with your hpp_key, or your merchant number and device number in the same email.

When using a Third-party Service/Shopping-cart Provider:

If your clients experience any technical difficulties, their first point of contact should be your Application Helpdesk. Once your Helpdesk agent determines that the problem resides on the Moneris side, the client and/or the agent can contact our Helpdesk above for support. Your product documentation should include this instruction and the contact information above.

If you require technical assistance while integrating your store, please contact the Moneris Gateway Integration Support Helpdesk below:

Phone: 1-866-562-4354

Fax: 416-734-1059

Email: eproducts@moneris.com

Hours: Monday – Friday, 8 am to 8 pm ET

When sending an email be sure to include your name and phone number as well as a clear description of the problem as well as the type of API that you are using. For security reasons, please do not send us your API Token via e-mail. Also, please do not send a combination of your store id or your merchant number and device number in the same email.

Table of Contents

About this documentation

This document contains a guide for using the Hosted Payment Page configuration tool in the Merchant Resource Center of Moneris Gateway. Also described are the methods for sending credit card transactions and managing the responses using Hosted Payment Page.

In addition to sending basic transaction information, this guide also provides information needed to implement security and fraud prevention tools via the Hosted Payment Page. To help prevent fraudulent activity on online transactions it is highly recommended that you also implement all of the other fraud tools available through Moneris Gateway:

- Address Verification Service (AVS) – Verifies the cardholder's billing address information.
- Card Validation Digit (CVD) – Validates that cardholder has a genuine credit card in their possession during the transaction.
- Verified by Visa (VbV), MasterCard SecureCode (MCSC), and American Express SafeKey – Authenticate the cardholder at the time of an online transaction.
- Transaction Risk Management Tool – Provides additional information and a risk score to help identify potential fraud.

This documentation contains the methods for sending a transaction and managing the response for an existing Hosted Payment Page user who wishes to add INTERAC® Online Payment functionality.

This document contains a guide for using the Hosted Tokenization configuration tool in the Merchant Resource Centre of Moneris Gateway. Also described are the methods for sending and processing a Hosted Tokenization transaction and managing the responses from the transaction.

This document contains a guide for using the Hosted Vault configuration tool in the Merchant Resource Centre of Moneris Gateway. Also described are the methods for sending and processing a Hosted Vault transaction and managing the responses from the transaction.

You have the ability to use other integration methods available through Moneris Gateway such as an API or Batch File Upload to process follow-on transactions such as Captures, Voids or Refunds which are not available using the Hosted Payment Page. The Merchant Resource Center can also be used to process such follow-on transactions.

Skills and System Requirements

In order to use Hosted Payment Page your system will need the following:

- A web server capable of sending and receiving an HTML POST/GET

In addition, you will need the following knowledge and/or skill set:

- Knowledge of creating an HTML web page and posting forms.
- Knowledge of iframes
- If you are selling more than one item, you will need some knowledge of a client-side scripting language (JavaScript, PHP, etc.) to calculate a final charge amount.
- If you want to create your own custom receipts and perform transaction verification, you will require knowledge of a server-side scripting language (PHP, Perl, ASP, etc.)

It is important to note that all Merchants and Service Providers that store, process, or transmit cardholder data must comply with PCI DSS and the Card Association Compliance Programs. However, certification requirements vary by business and are contingent upon your "Merchant Level" or "Service Provider Level". Failure to comply with PCI-DSS and the Card Association Compliance Programs may result in a Merchant being subject to fines, fees or assessments and/or termination of processing services. Non-compliant solutions may prevent merchants boarding with Moneris Solutions.

As a Moneris Solutions client or partner using this method of integration, your solution must demonstrate compliance to the Payment Card Industry Data Security Standard (PCI DSS) and/or the Payment Application Data Security Standard (PA DSS). These standards are designed to help the cardholders and merchants in such ways as they ensure credit card numbers are encrypted when transmitted/stored in a database and that merchants have strong access control measures.

For further information on PCI DSS and PA DSS requirements, please visit http://www.pcisecuritystandards.org.

For more information on how to get your application PCI-DSS compliant, please contact our Integration Specialists and visit https://developer.moneris.com to download the PCI-DSS Implementation Guide.

1 Introduction to Moneris Hosted Payment Solutions

Moneris' Hosted Payment Page Solution allows you as a merchant to securely accept payment information from your customers. This is one of the simplest integration methods to accept payments on your website. A few simple lines of coding will allow you to get started with your online payments.

1.1 Integrating Your Hosted Solution

To integrate a Moneris Gateway Hosted Solution, there are five overall sets of tasks and/or activities that you must complete:

1. **Creating a new configuration profile for the solution**. This is done in the Merchant Resource Center. For more information on this, see 2.1 Creating a Hosted Payment Solution Configuration Profile on page 13
2. **Configuring the solution profile**. Configuration is also done using the Merchant Resource Center's Hosted Payment Page Configuration Tool. For more information on this, see 2.2 Configuring a Hosted Payment Solution Profile on page 14
3. **Developing the solution**. For more information on this, see 3 Developing Your Hosted Solution on page 30.
4. **Testing the solution**. For more information on this, see 4 Testing a Hosted Payment Solution on page 70.
5. **Moving the solution into production**. For more information on this, see 5 Moving to Production on page 104

1.2 Hosted Payment Page

The Hosted Payment Page was designed as a solution for those merchants that do not wish to handle credit card information, or who do not have an SSL certificate.

When a transaction is initiated the transaction specific information is sent to Moneris Gateway Hosted Payment Page via an HTTP POST. The cardholder can then securely enter their credit card information. Upon submission Moneris Gateway will either generate a receipt on behalf of the merchant, or forward the cardholder and the response back to the merchant's site so that a custom receipt can be created. Upon receiving the response , the merchant can perform a Transaction Verification to ensure that the response is from a legitimate transaction.

1.3 INTERAC® Online Payment

INTERAC® Online Payment is offered to Canadian cardholders and allows them to perform transactions using their online banking. The INTERAC® Online Payment solution for the Moneris Gateway Hosted Payment Page (HPP) was designed so that very little extra integration would be necessary. If you are using Moneris Gateway-generated receipts no extra integration is required.

Current users of the Hosted Payment Page will notice that an extra screen appears before payment is accepted, this screen allows the cardholder to choose how they wish to pay and makes the payment process as straightforward as possible.

To enable INTERAC® Online Payment and to be able to accept INTERAC® Online Payment on your Hosted Payment Page, certification is required.

1.4 Convenience Fee

The Convenience Fee program was designed to allow merchants to offer the convenience of an alternative payment channel to the cardholder at a charge. This applies only when providing a true "convenience" in the form of an alternative payment channel outside the merchant's customary face-to-face payment channels. The convenience fee will be a separate charge on top of what the consumer is paying for the goods and/or services they were given, and this charge will appear as a separate line item on the consumer's statement.

NOTE: Convenience Fee does not necessarily work in combination with the other solutions in this section.

1.5 Transaction Risk Management Tool

The Transaction Risk Management Tool provides additional information to assist in identifying fraudulent transactions. For merchants who have enrolled in the Transaction Risk Management Tool, the Hosted Payment Page can be configured to obtain a risk assessment for every transaction and return the results along with the transaction response information.

1.6 Hosted Tokenization

The Moneris Hosted Tokenization (HT) was designed as a solution for online e-commerce merchants that do not wish to handle credit card information directly on their websites and also have the ability to fully customize their check-out webpage's appearance.

When HT is initiated, the Moneris Gateway will present and display on the merchant's behalf, a text-box on the check-out page for card number entry. Optionally, within the HT iframe, a text box for the expiration date and CVD data can be displayed as well along with the card number text box. The cardholder can then securely enter their credit card information into the text-boxes within the iframe.

Upon submission of the payment information on the check-out page, the Moneris Gateway will return a temporary token to the merchant, representing the credit card information. This token would then be used by the merchant to process a financial transaction via API with Moneris in order to charge the card.

Upon receiving a response to the financial transaction, the merchant would then generate a receipt and allow the cardholder to continue on with the online shopping experience.

A benefit of integration with Moneris' Hosted Tokenization is the solution will reduce a merchant's PCI-compliance assessment scope due to the fact that credit card information is not captured nor stored by the merchant's site.

1.7 Hosted Vault Account Registration & Update (HVARU)

The Hosted Vault Account Registration & Update Page (HVARU) was designed as a solution for those merchants that do not wish to handle client's credit card or account information. When a transaction is initiated, the transaction specific information is sent to Moneris Gateway HVARU via an HTTP POST. The cardholder can then securely enter their account information. Upon submission Moneris Gateway will forward the cardholder and the response back to the merchant's site so that further action can be taken by the merchant website. Also, upon receiving the response the merchant can perform a Transaction Verification to ensure that the response is from a legitimate request.

The Hosted Vault Account Registration & Update Page will not send any financial transactions — it is used to register and update account information only. The data returned from the HVARU account registration and update is to be used in conjunction with the Vault API of your choice. The APIs are capable of sending financial transactions, and updating account information. For PCI reasons some merchant accounts may be unable to alter the financial account portions of the account through the API; however, updating all other parts of a registered account is possible.

1.8 Gift Cards

Gift Card payments are now supported in the Hosted Payment Page. If your merchant account supports gift cards and gift cards are enabled in the Hosted Payment Page configuration the cardholder will be given the opportunity to pay using gift cards. Each Gift Card transaction is limited to two gift cards and one other form of payment (credit card or INTERAC® Online Payment).

Due to the possible complexity of the gift card response, there are only two options for handling the transaction response: Moneris Gateway generates the receipt or the response is returned in XML format. If data is being returned in XML, then all data must be displayed on the receipt, with the exception of the txn_num and result.

1.9 Loyalty Cards

Loyalty Cards are now supported in the Hosted Payment Page. If your merchant account supports loyalty cards and Redeem Loyalty Points and/or Award Loyalty Points are enabled in the Hosted Payment Page configuration the cardholder will be given the opportunity to provide their loyalty card number.

Due to the possible complexity of the loyalty card response there are only two options for handling the transaction response: Moneris Gateway generates the receipt or the response is returned in XML format. If data is being returned in XML all data must be displayed on the receipt with the exception of the txn_num and result. Please refer to Appendix E XML POST Response for Financial Transaction on page 111

2 Hosted Payment Page Configuration Tool

- 2.1.1 Creating a Hosted Payment Page Configuration
- 2.1.2 Creating a Hosted Tokenization Configuration
- 2.1.3 Creating a Hosted Vault Configuration
- 2.2.1 Configuring the Hosted Payment Page
- 2.2.2 Configuring the Hosted Vault Page

The Hosted Payment Page Configuration Tool, part of the Merchant Resource Center, is where you create and configure a profile for your hosted payment solution. Creating and configuring a hosted payment solution profile are the two first steps in the process of integrating the hosted payment solution with your system. To review the steps for integrating your hosted payment solution, refer to Integrating Your Hosted Solution (see page 9).

In order to use the Hosted Payment Page Configuration Tool to create and configure a hosted payment solution profile, you need to log in to the Merchant Resource Center test environment.

To log into the Merchant Resource Center test environment go to

https://esqa.moneris.com/mpg

and use one of the following login IDs.

Table 1: Test IDs for Merchant Resource Center

Store ID	Username	Password
store1	DemoUser	password
store2	DemoUser	password
store3	DemoUser	password
store5	DemoUser	password
moneris	DemoUser	password

Table 2: Test IDs for Merchant Resource Center - INTERAC® Online Payment

Store ID	Username	Password
store3	DemoUser	password

Table 3: Test IDs for Merchant Resource Center - Convenience Fee

Store ID	Username	Password
monca00392	DemoUser	password

2.1 Creating a Hosted Payment Solution Configuration Profile

The first step in the process of integrating your hosted payment solution is creating a configuration profile for it using the Merchant Resource Center Hosted Payment Page Configuration Tool.

2.1.1 Creating a Hosted Payment Page Configuration

To create a Hosted Payment Page Configuration:

1. Log in to the Merchant Resource Center
   - QA: https://esqa.moneris.com/mpg
   - Production: https://www3.moneris.com/mpg
2. Click on Admin in the menu
3. Click **Hosted Paypage Config**
4. Click the **Generate a New Configuration** button
5. Complete your configuration as outlined in 2.2.1 Configuring the Hosted Payment Page on page 15

You will be assigned a Hosted Payment Page ID (ps_store_id) this is the identifier for this unique configuration. You will also be assigned a Hosted Payment Page Token (hpp_key). The Hosted Payment Page ID and Token are sent as part of the transaction request to securely identify your store and the specific configuration.

Each Moneris Gateway account may have up to five unique Hosted Payment Page configurations. These do not equate to different stores — all transactions will be logged under the same store and will settle into the same bank account. Each configuration can have a different appearance as well as handle responses in varying ways. Because the Moneris Gateway test environment is a shared environment there is no limit to the number of configurations assigned to a specific store account. However, there is a 30 day time limit where a store configuration will be deleted after 30 days, regardless of use. Please do not alter or delete configurations that were not generated by you.

NOTE: In the production environment, an individual must be granted permission to access and alter the configuration.

2.1.2 Creating a Hosted Tokenization Configuration

To create a Hosted Tokenization Configuration:

1. Log in to the Merchant Resource Center
    - QA: https://esqa.moneris.com/mpg
    - Production: https://www3.moneris.com/mpg
2. Click on **Admin** in the menu
3. Click on **Hosted Tokenization** in the sub-menu
4. (Optional) Enter the Source Domain page. This is the address of the main outer page that sends the transaction to the Moneris Gateway. Example from the process flow diagram above would be “https://www.xyz.com”. This may be left blank for mobile solutions or if the profile is being used by multiple domains.
5. Click **Create Profile**
6. Make a note of the Profile ID that is generated since this will need to be included in your HTML iframe code

2.1.3 Creating a Hosted Vault Configuration

To create a Hosted Vault Configuration:

1. Log in to the Merchant Resource Center
    - QA: https://esqa.moneris.com/mpg
    - Production: https://www3.moneris.com/mpg
2. Do the required development as outlined in "Hosted Vault" on page 1
3. Click on Vault in the menu
4. Click **Hosted Vault Config**
5. Click the **Generate a New Configuration** button

You will be assigned a Hosted Vault ID (res_id), which is the identifier for this unique configuration. You will also be assigned a Hosted Vault Token (res_key). The Hosted Vault ID and Token are sent as part of the registration/update request to securely identify your store and the specific configuration.

Each Moneris Gateway account may have up to five unique Hosted Vault configurations. Each configuration can have a differing appearance as well as handle responses in varying ways. Because the Moneris Gateway test environment is a shared environment there is no limit to the number of configurations assigned to a specific store account. However, there is a 30 day time limit where a store configuration will be deleted after 30 days, regardless of use. Please do not alter or delete configurations that were not assigned to you.

NOTE: In the production environment, an individual must be granted permission to access and alter the configuration.

2.2 Configuring a Hosted Payment Solution Profile

Once you have created a profile for your hosted payment solution, the next step is to configure that profile using the Merchant Resource Center Hosted Payment Page Configuration Tool.

2.2.1 Configuring the Hosted Payment Page

Generate new hpp_key

This allows you to change the Hosted Payment Page Token (hpp_key). Both the "ps_store_id" and "hpp_key" are to be kept secure, though if security were to be compromised, you may generate a new "hpp_key" without having to create a completely new configuration.

> **NOTE:** After clicking **Generate new HPP key**, your current key will cease to work immediately — there is also no way to retrieve or revert to the old key.

Basic Configuration

Description

Add a description to easily identify this Hosted Payment Page configuration. This is especially useful when maintaining more than one Hosted Payment Page configuration.

Transaction Types

This defines what type of transaction will be processed.

Purchase: The cardholder will be charged immediately and funds will be deposited next business day. This is used if your goods and services are shipped/provided within 24 hours.

Preauthorization: The funds will be locked but will not be settled until a Capture is performed. The Capture will need to be performed via the Merchant Resource Center or via an API. Preauthorization (PreAuth) is used if the goods and services are not shipped/provided within 24 hours.

Payment Methods

This defines which Payment Methods the Hosted Payment Page will allow, for example, one or all of the methods listed below. The Payment Methods available depend on the store's set-up and these may include Credit Cards, INTERAC® Online Payment or Gift Cards.

Credit Card: This will offer the customer the option of paying with their credit card once they reach the Hosted Pay Page. Please note, when the customer chooses to pay with a credit card, as opposed to INTERAC Online or Gift Cards, the transaction type that will be processed will be the one defined in the default 'Transaction Type' section above.

INTERAC® Online Payment: This will offer the customer the option of being forwarded to their online banking to approve the transaction. The funds will be debited directly from their bank account. Please note, all INTERAC® Online Payment transactions using the Hosted Payment Page will be processed as purchase transactions, there are also unique receipt requirements for INTERAC® Online Payment transactions.

Gift Cards: This will offer the customer the option of using up to two gift cards as part of the transaction. All gift card transactions will be processed as purchase transactions. The customer will have the option to check their gift card balance on the Hosted Payment Page.

MasterPass: This option enables MasterCard cardholders to use the information in their MasterPass digital wallets. The customer will be forwarded to their MasterPass wallet site when they choose to use this payment method.

Visa Checkout: This option enables Visa cardholders to use the information in their Visa Checkout digital wallets. The customer will be forwarded to their Visa Checkout wallet site when they choose to use this payment method.

Loyalty Cards: This will allow the cardholder the opportunity to provide their loyalty card number to redeem Loyalty Points and/or earn Loyalty Points

Response Method

This determines how the transaction response will be handled.

Moneris Gateway will generate a receipt: Once the transaction is processed Moneris Gateway will generate and display a receipt page based on the Pay Page Appearance and Response Data configurations.

Sent to your server as a POST: Moneris Gateway will use an HTTP POST to send the transaction responses to your web server so that you can customize the receipt or so that otheCor processes may be initiated on your site.

Sent to your server as a POST containing XML: Moneris Gateway will use an HTTP POST to send the transaction responses back to your web server so that you can customize the receipt or so that other processes may be initiated on your site. The response will be in XML format that will need to be parsed.

Sent to your server as a GET: Moneris Gateway will redirect the cardholder to a URL on your server and attach the response as a URL encoded query string at the end of the URL so that you can customize the receipt or so that other processes may be initiated on your site. Please note that there are limitations imposed by the browser and operating system on the length of a query string.

> **NOTE:** When handling the response (POST or GET), you must be able to dynamically parse the data. In the future, new variables may be added and the order of the response variables may change.

Approved URL

If you have chosen to create your own receipt (in Response Method) you will need to specify the URL where the transaction response will be returned when it is approved. All URLs need to be complete, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

Declined URL

If you have chosen to create your own receipt (in Response Method) you will need to specify the URL where the transaction response will be returned when it is declined. This can be the same as the Approved URL. All URLs need to be complete, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

> **NOTE:** Click on the **Save Changes** button to save the existing configuration. If the configuration is not saved the current Hosted Payment Page ID (ps_store_id) and Token (hpp_key) will be deleted after approximately 15 minutes.

Enhanced Cancel

If you have chosen to use the Enhanced Cancel feature, a full response will be returned to the Declined URL with a response code of '914' and a message of 'cancelled by cardholder'.

Payment Page Appearance

> **NOTE:** Click on the **Save Appearance Settings** button to apply these appearance settings to the Hosted Payment Page. If the appearance configuration is not saved these fields will be returned to their last known saved configuration.

Colours and Styles

The following fields define the colour scheme that is to be used for the Hosted Payment Page as well as the receipts. The colours must be defined in 6 character hex – there is a hex colour chart available by clicking on **Hex Colour Chart** button.

Background Colour

This defines the background colour of the page.

Font Style

This defines what font group will be used for the Hosted Payment Page. We have defined three groups – Arial/Helvetica/SansSerif, Times New Roman/Times/Serif and Courier New/Courier/Mono.

Primary Text Colour

This defines the colour for the majority of the text on the colour. This must be legible on the chosen background colour.

Company Name Colour

This defines the colour that will be used for your company name.

Header and Footer Highlight Colour

This defines the colour that will be used for a highlight bar that will appear below the company name and at the bottom of the Hosted Payment Page.

Section Divider Colour

The Hosted Payment Page will be divided into several sections depending on what is displayed. A colour bar is used to define the information. This defines the colour of the Primary Bar.

Section Divider Text Colour

Occasionally the primary colour bar will contain text — this defines the colour of the text that will appear in the Primary Bar. Please ensure that the text is legible.

Subsection Divider Colour

The Hosted Payment Page can be divided into several sub-sections depending on what is displayed. A colour bar is used to define the information. This defines the colour of the secondary bar that may subdivide information.

Subsection Divider Text Colour

Occasionally the Subsection Divider will contain text — this defines the colour of the text that will appear in the Subsection Divider. Please ensure that the text is legible.

Hosted Payment Page Data Fields

The following "Display" fields define what is to be displayed on the Hosted Payment Page where the cardholder enters their card information. Some fields are required to be sent by the merchant, others can be set as input boxes on the Hosted Payment Page. We do not check for completion or validity of the information input therefore we suggest that it be passed from the merchant to us.

Display item details

This field indicates whether a listing of items purchased, taxes and shipping costs is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Display customer details

This includes several supplemental data fields that the merchant may pass to the Hosted Payment Page such as a Customer Id, a Customer Email Address, and an additional data field referred to as the Note data.

Display billing address details

This field indicates whether the client's billing information is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Display shipping address details

This field indicates whether the client's shipping information is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Disable Address Input Boxes

This disables the input boxes so that the cardholder cannot alter/add information in the Address, Note and Email fields. When the input boxes are disabled the data will appear as text.

Display Merchant Name

This field indicates whether the Merchant Name should also be displayed on the Hosted Payment Page. The name that will be displayed is the official Merchant Name that Moneris Solutions has associated with the account and the name that the cardholder will see on their credit card statement. It is mandated by industry regulations that the merchant name be displayed on any checkout pages and receipts, but this field may be omitted if the Hosted Payment Page will be loaded within a frame that already displays the merchant name. If you choose to load the Hosted Payment Page within a frame, you are then required to have an SSL certificate.

Buttons

Cancel Button Text

This configures the text that is to appear on the Cancel button. The Cancel button appears on the credit card input page and allows the cardholder to return to your site if they do not wish to complete the transaction.

Cancel Button URL

This configures the URL associated with the Cancel button. The Cancel button appears on the credit card input page and allows the cardholder to return to your site if they do not wish to complete the transaction. All URLs need to be complete, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

Continue Button Text

This configures the text that is to appear on the continue button. The continue button appears on the receipt page that is generated if the option in “Response Methods” is chosen, otherwise this field will be hidden. The Continue button allows the client to return to your site after completing a transaction.

Continue Button URL

This configures the URL associated with the Continue button. The Continue button appears on the receipt page that is generated if the option in “Response Methods” is chosen, otherwise this field will be hidden. All URLs need to be complete, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

Hide Continue Button on Receipt Page

This configures whether the Continue button is to appear in the receipt page or not. We do not advise hiding the Continue button.

Hosted Payment Page Input Fields

The following section defines what input fields are to be displayed on the Hosted Payment Page. When an input field is included, it will automatically be mandatory for the customer to fill it in.

Display CVD input (Credit Card only)

This defines whether the Hosted Payment Pageshould include the prompt for the Card Validation Digits (CVD). If this input field is displayed on the Hosted Payment Page it is then mandatory that the cardholder complete this data. This input field only applies to MC, VISA, AmEx and other Credit Card plans that support CVD transactions.

Display AVS input (Credit Card only)

This defines whether the Hosted Payment Page should include the prompt for the Address Verification Service (AVS) details. If these input fields are displayed on the Hosted Payment Page it is then mandatory that the cardholder complete this data. These input fields only apply to MC, VISA, AmEx and other Credit Card plans that support AVS transactions.

Logos

This section defines what logos will appear on the payment page. Please note that credit card logos are for display only and will not affect what card types you are able to accept. To begin accepting a new card type, please contact the Service Centre at 1-866-319-7450.

Credit Card Logos

Allows you to select which logos should appear on the Hosted Payment Page. Check off the appropriate logos: Visa, MasterCard, American Express, Diners, Discover, Sears, JCB, Visa Debit.

Response/Receipt Data

Click on the **Configure Response Fields** button to specify what additional fields you wish to have returned in the transaction response or displayed on the response receipt.

New to this version is the ability to pass back line items, shipping and billing data that previously had to be passed in as “rvar” in order to be returned to the merchant. We still strongly advise storing customer details on the merchant’s server before passing them to the Hosted Payment Page.

> **NOTE:** Click on the **Save Response Settings** button to apply these chosen additional fields to the transaction response. If the response configuration is not saved these fields will be returned to their last known saved configuration. Next, click on the **Return to main configuration** button to continue with your Hosted Payment Page configuration setup.

Response/Receipt Field Configuration

Return line item details

All line item details will be returned to the response URL in the same manner they were passed to the Hosted Payment Page in the request.

Return shipping details

All shipping details will be returned to the response URL in the same manner they were passed to the Hosted Payment Page in the request.

Return billing details

All billing details will be returned to the response URL in the same manner they were passed to the Hosted Payment Page in the request.

Return other customer fields

Fields such as cust_id, email, and note will be returned to the response URL in the same manner they were passed to the Hosted Payment Page in the request.

Return ECI value

The ECI value that was used during transaction processing will be returned to the response URL. This is used to determine the result of a VbV/MCSC transaction. It is strongly encouraged to review all orders even when it appears a VbV/MCSC authentication was successful or attempted.

> **NOTE:** The eci value sent in a follow-on transaction request must reflect the eci value received from the Preauthorization.

EXAMPLE: if the eci value received from the preauthorization is "5", then the subsequent API capture request must also have its eci set to “5”.

Return the txn_number

The txn_number for the transaction number is returned in the response. This allows automation of captures, voids and refunds through the use of an API.

Return the VbV Result Code

The VbV result code value from Visa will be returned to the response URL. This is used to determine the validity of the VbV transaction data. It is strongly encouraged to review this for all VbV transactions.

Return a Visa Debit card indicator

A value of true or false is sent back indicating if the card provided by the cardholder was a Visa Debit card.

Return AVS data

The Address Verification data entered by the cardholder on the Hosted Payment Page will be returned to the response URL.

Asynchronous Transaction Response

Perform asynchronous data post

This must be checked for asynchronous data post to be enabled. When the asynchronous data post is enabled the Hosted Payment Page will perform a server to server post of the response data as a secondary method of getting the response data. This does not replace the normal transaction response which will still be sent through the browser as a POST or a GET. This is supplementary and can be used to verify/validate the browser response.

If you have enabled the asynchronous data post within your production Hosted Payment Page, you will need to specify the response URL in (HTTPS) format. Self signed certificates will be accepted, but an HTTP address will not work.

NOTE:

If you send characters that are not supported in any of the variables, the extra transaction details may not be stored, displayed or returned in the response.

The order_id allows the following characters: a-z A-Z 0-9 _ - : . @ spaces

All other request fields allow the following characters: a-z A-Z 0-9 _ - : . @ $ = /

If you are using accents they must be sent as html entities (é = &eacute;) If these are being sent as GET please note that they must be URL encoded.

Also, please note that if the response is to be sent as a GET the extra transaction details may not always be properly returned. This is due to limitations imposed by the browser and operating system on the length of a query string.

Security Features

Click **Configure Security** to add extra security measures to the Hosted Payment Page.

Referring URL

By adding a URL, you specify that you would like us to check whether the transaction is coming from a location (URL) that you allow. Only POSTs sent from one of the specified URLs will be processed. (It is possible for the Referring URL to be “spoofed” – this is not a guaranteed method of securing your transactions – but it makes it more difficult).

Add URL

Here you can specify up to ten Referring URLs to a max of 255 characters. Each URL needs to be complete and at a registered domain – www.example.com is not sufficient – the proper URL is http://www.example.com/index.html (IP addresses are not supported). After specifying a URL, click on the **Add URL** button to add it to the Allowed URLs list. Once a URL has been added, the **Remove URL** button will become available.

> **NOTE:** To verify your Referring URL, you may POST to https://esqa.moneris.com/mpg which will display the URL you are posting from.

Transaction Risk Scoring

Enable Transaction Risk Scoring

This must be checked for transaction risk scoring to be enabled. When Transaction Risk Scoring is enabled the Hosted Payment Page will automatically generate a “session_id” in the transaction request and send this to ThreatMetrix for device profiling. When the response is received, Moneris Gateway then replies with the transaction information and the transaction risk score. Each transaction can only be verified once. The response to the Transaction Risk Scoring will be sent to your server in the form of a POST or GET only.

Card Verification

Enable Card Verification

This must be checked for a card verification transaction to be performed. When Card Verification is enabled the Hosted Payment Page will check the validity of a credit card before it is registered for recurring billing. This will only be performed when 'Bill Now' is set to 'false'.

Transaction Verification

Enable Transaction Verification

This must be checked for transaction verification to be enabled. When Transaction Verification is enabled the Hosted Payment Page will return a “transactionKey” in the transaction response. When the response is received the fields should be logged and a transaction verification request is sent to Moneris Gateway. Moneris Gateway then replies with transaction information and whether the transaction was valid or not. Each transaction can only be verified once and it must be verified within 15 minutes of the original transaction being performed. This allows you to ensure that the responses sent to your page are not “spoofed” and that you are only receiving the responses once. If you also intend to check the Referring URL you must ensure that the source of the verification request is in the list of Allowed URLs.

Response Method

This determines how the transaction verification response will be handled.

Sent to your server as a POST: Moneris Gateway will use an HTTP POST to send the transaction verification responses to your web server so that other processes may be initiated on your site.

Sent to your server as a GET: Moneris Gateway will redirect the cardholder to a URL on your server and attach the transaction verification response as a URL encoded query string at the end of the URL so that other processes may be initiated on your site.

Displayed as XML on our server: Once the transaction verification has been performed Moneris Gateway will generate a page and display an XML string. This can be used in conjunction with cURL, screen scraping or other such methods.

Displayed as key/value pairs on our server: Once the transaction verification has been performed Moneris Gateway will generate a page and display key value pairs. This can be used in conjunction with cURL, screen scraping or other such methods.

NOTE: When handling the response, you must be able to dynamically parse the data. In the future, new variables may be added and the order of the response variables may change.

Response URL

If you have chosen to have the transaction verification response sent back to you in either a POST or GET (in Response Method) you will need to specify the URL where the transaction response will be returned. The URL needs to be complete and at a registered domain, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

If you have chosen to have Moneris Gateway display an XML string or key/value pairs this field may be left blank.

NOTE: Click on the **Save Verification Settings** button to apply these chosen additional security features to the Hosted Payment Page. If the security feature is not saved these fields will be returned to their last known saved configuration. Next, to continue with your Hosted Payment Page configuration setup, click the **Return to main configuration** button.

Email Receipts

Click on **Configure Email Receipts** to specify email receipt conditions and appearance.

NOTE:

New to this version: All emails are sent in HTML and text format. This change will allow recipient's email client to display their receipt in their default preferred format. This will allow better formatting for customer receipts which will accommodate both web and mobile receipts.

Although we are sending in both HTML and text format, the cardholder will only receive one receipt.

All emails now come from the www.moneris.com mail server vs. the www3.moneris.com mail server which had been identified as being a problem for certain spam filters.

Receipt Conditions

Send email to cardholder if transaction approves

This defines whether a receipt is to be e-mailed to the cardholder if the transaction approves. If this option is selected, but the cardholder's e-mail address is not sent in the POST, then the receipt will not be emailed out. Please refer to Other Details for Basic Transaction (see page 34) for an example of how to send the cardholder's e-mail address ("email") in the request.

Send email to cardholder if transaction declines

This defines whether a receipt is to be e-mailed to the cardholder if the transaction declines. If this option is selected, but the cardholder's e-mail address is not sent in the POST, then the receipt will not be emailed out. Please refer to Other Details for Basic Transaction (see page 34) for an example of how to send the cardholder's email address ("email") in the request.

Send email to merchant if transaction approves

This defines whether a receipt is to be e-mailed to the merchant if the transaction approves. If this option is selected, but the merchant's e-mail address is not provided in the "Merchant email address" field, then the receipt will not be emailed out.

Send email to merchant if transaction declines

This defines whether a receipt is to be e-mailed to the merchant if the transaction declines. If this option is selected, but the merchant's e-mail address is not provided in the "Merchant email address" field, then the receipt will not be emailed out.

Include 'rvar' in merchant email

"rvar"s can be a series of variables/values that will be echoed back in the transaction response. This field indicates whether these fields are to be included in the email to the merchant. They will NOT be included in the email to the client. If this option is selected, but the merchant's e-mail address is not provided in the "Merchant email address" field, then the "rvar"s will not be e-mailed to the merchant.

Merchant email address

This field allows you to provide the e-mail address you want all the Merchant Email Receipts to be sent to, as defined in the 3 options mentioned above. Only one e-mail address may be provided.

Receipt Appearance

Include Line Item Details

This field indicates whether a listing of items purchased, taxes and shipping costs is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Include Billing Details

This field indicates whether the client's billing information is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Include Shipping Details

This field indicates whether the client's shipping information is to be displayed. In order to display this information it is required that it be sent in the transaction request. Please see request variables to properly send this data.

Include Customer Details

This will include the cust_id, client email address, and the note field data.

Email Text

New to version 3 of the Hosted Payment Page is the ability to add a short message that will appear at the top of the email receipts. There is a 255 character limit and the characters supported are limited to letters, numbers and the following characters ' # @ _ , - . and space and enter (newline).

NOTE:

If you send characters that are not supported in any of the variables, the extra transaction details may not be stored or included in the email receipt.

The order_id allows the following characters: a-z A-Z 0-9 _ - : . @ spaces

All other request fields allow the following characters: a-z A-Z 0-9 _ - : . @ $ = /

NOTE:

Click the **Save Email Settings** button to apply these chosen Email Receipt settings to the Hosted Payment Page. If the Email Receipt configuration is not saved these fields will be returned to their last known saved configuration. Next, to continue with your Hosted Payment Page configuration setup, click the **Return to main configuration** button .

2.2.2 Configuring the Hosted Vault Page

Generate new res_key

This allows you to change the Hosted Vault Token (res_key). Both the "res_id" and "res_key" are to be kept secure, though if security were to be compromised, you may generate a new "res_key" without having to create a completely new configuration.

Basic Configuration

Response Method

This determines how the transaction response will be handled.

Sent to your server as a POST: Moneris Gateway will use an HTTP POST to send the transaction responses to your web server so that you can store the data_key and proceed with other processes which may be associated with account creation or updates.

Sent to your server as a GET: Moneris Gateway will redirect the cardholder to a URL on your server and attach the response as a URL encoded query string at the end of the URL so that you can store the data_key and proceed with other processes which may be associated with account creation or updates. Please note that there are limitations imposed by the browser and operating system on the length of a query string.

> **NOTE:** When handling the response (POST or GET), you must be able to dynamically parse the data. In the future, new variables may be added and the order of the response variables may change.

Response URL

You will need to specify the URL where the transaction response will be returned (either in a POST or GET). The URL needs to be complete , for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

> **NOTE:** Click on the **Save Changes** button to save the existing configuration. If the configuration is not saved the current Hosted Vault ID (res_id) and Token (res_key) will be deleted after a period of time

Hosted Vault Page Appearance

Click on the **Configure Appearance** button to specify what will be displayed on the Hosted Payment Page.

Hex Colour Chart

All colours in the Colours and Styles section must be input as the standard HTML hex colour value. You may click this button to view a colour chart.

Colours and Styles

Background Colour

This defines the background colour of the page.

Font Style

This defines what font group will be used for the HVARU. We have defined three groups – Arial/Helvetica/SansSerif, Times New Roman/Times/Serif and Courier New/Courier/Mono.

Primary Text Colour

This defines the colour of text on the HVARU. This must be legible on the chosen background colour.

Company Name Colour

This defines the colour that will be used for your company's business/merchant name.

Header & Footer Highlight Colour

The HVARU will contain a header and footer and a colour bar is used to define these sections. This defines the colour of the header and footer bars.

Section Divider Colour

The HVARU may be divided into several sections depending on what is displayed. A colour bar is used to define the information. This defines the colour of the section divider bars.

Section Divider Text Colour

Occasionally the Section Divider Bar will contain text – this defines the colour of the text that will appear in the Section Divider Bar. Please ensure that the text is legible (i.e. Do not pick the same colour for the Section Divider Colour and for the Section Divider Text Colour).

Subsection Divider Colour

The HVARU may be divided into several sub-sections depending on what is displayed. A colour heading highlight is used to define the information. This defines the colour of the sub-section headings.

Subsection Divider Text Colour

Occasionally the sub-section headings will contain text – this defines the colour of the text that will appear in the Subsection Divider. Please ensure that the text is legible (i.e. Do not pick the same colour for the Subsection Divider Colour and for the Subsection Divider Text Colour).

Hosted Vault Page Data Fields

Display customer details (cust_id, email, note, phone . . .)

This field indicates whether the Customer ID ("cust_id") is to be displayed on the HVARU – the Customer ID field is often used for membership #'s, policy #s, student IDs, or invoice #s. It is a searchable field from the Merchant Resource Center. Also, this field will indicate whether the HVARU should display other fields such as the customer's email address ("email"), the phone number ("phone") and the "note" field – the "note" field can contain any special instructions. In order to display this information it is required that the "cust_id", "email", "phone" and "note" fields be sent in the transaction request. Please refer to Required Variables - Adding New Profile to Vault (see page 62) to properly send this data.

Display merchant name

This field indicates whether the Merchant Name should also be displayed on the HVARU. The name that will be displayed is the official Merchant Name that Moneris Solutions has associated with the account and the name that the cardholder will see on their credit card statement. It is mandated by industry regulations that the merchant name be displayed on any checkout pages and receipts, but this field may be omitted if the HVARU will be loaded within a frame that already displays the merchant name. If you choose to load the HVARU within a frame, you are then required to have an SSL certificate.

Hosted Vault Page Input Fields

Display AVS input

This defines whether the HVARU should include the prompt for the Address Verification Service (AVS) details. If these input fields are displayed on the Hosted Paypage it is then mandatory that the cardholder complete this data.

Security Features

Referring URL

By adding a URL, you specify that you would like us to check whether the transaction is coming from a location (URL) that you allow. Only POSTs sent from one of the specified URLs will be processed. (It is possible for the Referring URL to be "spoofed" – this is not a guaranteed method of securing your transactions – but it makes it more difficult).

Add URL

Here you can specify up to ten Referring URLs to a max of 255 characters. Each URL needs to be complete and at a registered domain, for example:

not sufficient URL: www.example.com

proper URL: http://www.example.com/response.php

If a URL is missing or improperly typed it may result in a 404 error or a looping page.

After specifying a URL, click on the **Add URL** button to add it to the Allowed URLs list. Once a URL has been added, the **Remove URL** button will become available.

> **NOTE:** To verify your Referring URL, you may POST to https://esqa.moneris.com/HPPDP/myurl.php which will display the URL you are posting from.

Transaction Verification

> **NOTE:** Click on the **Save Verification Settings** button to apply these chosen additional security features to the Hosted Vault configuration. If the security feature is not saved these fields will be returned to their last known saved configuration.

Enable Transaction Verification

This must be checked for transaction verification to be enabled. When Transaction Verification is enabled the HVARU will return a "transactionKey" in the transaction response. When the response is received the fields should be logged and a transaction verification request is sent to Moneris Gateway. Moneris Gateway then replies with transaction information and whether the transaction was valid or not. Each transaction can only be verified once and it must be verified within 15 minutes of the original transaction being performed. This allows you to ensure that the responses sent to your page are not "spoofed" and that you are only receiving the responses once. If you also intend to check the Referring URL you must ensure that the source of the verification request is in the list of Allowed URLs.

Response Method

This determines how the transaction verification response will be handled.

Displayed as XML on our server: Once the transaction verification has been performed the HVARU will generate a page and display an XML string. This can be used in conjunction with cURL, screen scraping or other such methods.

Displayed as key/value pairs on our server: Once the transaction verification has been performed the HVARU will generate a page and display key value pairs. This can be used in conjunction with cURL, screen scraping or other such methods.

> **NOTE:** When handling the response (POST or GET), you must be able to dynamically parse the data. In the future, new variables may be added and the order of the response variables may change.

Card Verification

Enable Card Verification

This must be checked for card verification to be enabled. This allows our system to perform card verification on the card before adding the card to the vault. If the card verification fails, card is not added to the vault and no data key is returned.

> **NOTE:** Card verification will only be performed for VISA and MasterCard. All other card types are not supported.

Vault Update Settings

This section will allow you to have the update information page locked for a specified number of minutes after a set number of sequential failed key attempts have been made. An email can also be sent when the page is locked.

> **NOTE:** Click on the **Save Vault Update Settings** button to apply these chosen additional security features to the Hosted Vault configuration. If the security feature is not saved these fields will be returned to their last known saved configuration. Next, to continue with your Hosted Vault configuration setup, click the **Return to main configuration** button.

Number of attempts

This is the number of sequential failed attempts that the HVARU will allow before locking and preventing all future attempts for the number of minutes defined under Lock Period. For example, to lock the update functionality for 15 minutes after it has received 4 invalid data keys sequentially, please set the Number of Attempts to 4 and the Lock Period to 15.

Lock Period

This defines the number of minutes the update functionality will be locked. To lock the update functionality indefinitely set Lock Period to '999'.

Email Address

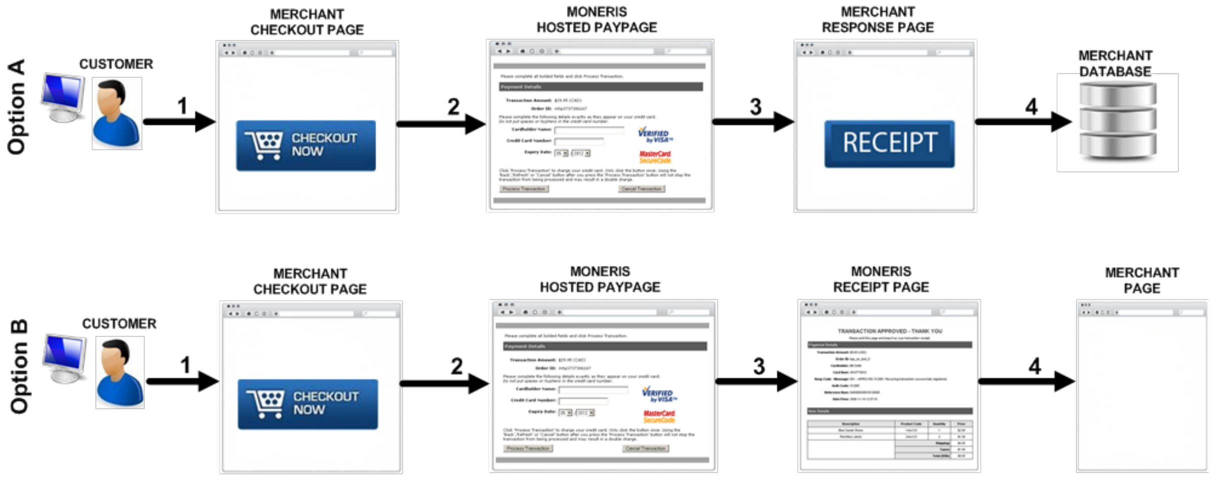
If the HVARU is locked, a notification email will be sent to this address if it is filled in. To receive an email after every invalid data key but to never lock the update feature, configure Number of Attempts to '1' and Lock Period to '0', and fill in Email Address.

> **NOTE:**
>
> DO NOT USE REAL ACCOUNT INFORMATION WHEN TESTING IN THE QA ENVIRONMENT.
>
> Moneris GatewayQA is a shared environment and data sent to it may be accessible to others.

3 Developing Your Hosted Solution

3.1 Developing for Your Hosted Payment Page

3.1.1 Hosted Payment Page Process Flow



Outlined above is the basic Hosted Payment Page flow which allows a cardholder to process a transaction. There are 2 options available:

A. This option is for merchants that choose to have the response sent back to them in either a POST or GET format so that they may build the receipt themselves, as well as store the response variables as needed.
B. This option is for merchants that choose to have Moneris generate the receipt.

The steps required to achieve this are as follows. Steps 1 and 2 are common to both options:

1. The customer arrives on the merchant's website (application). At this point the merchant must determine the amount of the transaction and also collect any additional data.
2. Once the cardholder is ready to pay, the merchant's checkout page will submit an HTTP form POST to the Moneris Hosted Payment Page (HPP). At this time, the customer will be redirected from the merchant's website to the Moneris Hosted Payment Page URL. For details on the HTTP form POST, please refer to "Performing a Transaction Verification" (page 44) of this document which outlines the mandatory and optional fields that may be sent to the Hosted Payment Page.

Then, follow steps 3 and 4 for the option you choose, as follows:

Option A

3. On the Hosted Payment Page the cardholder will fill in their secure payment details such as their card number or and submit the transaction. At this time, Moneris Gateway will process the transaction and then build a response to send back to the merchant in a POST or GET format to the Response URL provided in the Hosted Payment Page Configuration as described in Hosted Payment Page Configuration Tool (see page 12).
4. Once the merchant receives the response details they must provide a receipt to the customer and then may store these details for future purposes such as reporting or tracking.

Option B

3. On the Hosted Payment Page, the cardholder will fill in their secure payment details such as their card number and submit the transaction. At this time, Moneris Gateway will process the transaction and then display a receipt to the customer.
4. Once the customer is ready to continue they will then be redirected back to the merchant's website to Response URL provided in the Hosted Payment Page Configuration as described in Hosted Payment Page Configuration Tool (see page 12).

3.1.2 Sending a Transaction to the Hosted Payment Page

- 3.1.2.1 Basic Transaction to the Hosted Payment Page
- 3.1.2.2 INTERAC® Online Payment Transaction
- 3.1.2.3 Performing a Transaction Risk Management Tool Transaction

3.1.2.1 Basic Transaction to the Hosted Payment Page

Without any further customization, the Hosted Payment Page has a default set of transaction options which we refer to as "basic transactions." Below are a series of tables containing all the fields that can be sent in a Hosted Payment Page request for a basic transaction. The first table contains the required variables – these must be sent. Subsequent tables contain variables that can be sent optionally.

NOTE: While the use of basic transaction options only is acceptable, Moneris Solutions recommends that you also use preload and asynchronous response to enhance the security of the Hosted Payment

Page. For more information on these subjects, see "Data Preload with Hosted Payment Page" (page 47)and "Asynchronous Transactions in Hosted Payment Page" (page 53)

Required Variables for Basic Transaction

Table 1: Required Variables - Basic Transaction

Variable name	Type	Description
	form	https://esqa.moneris.com/HPPDP/index.php - Development https://www3.moneris.com/HPPDP/index.php - Production
ps_store_id	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool. Identifies the configuration for the Hosted Payment Page.
hpp_key	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool. Identifies the configuration for the Hosted Payment Page.
charge_total	hidden	Final purchase Amount - no $, must include 2 decimal places (i.e. 3.00) This amount must be higher than the Convenience Fee Amount

Sample with Required Variables

EXAMPLE:

Below is a sample of the most basic Hosted Payment Page call using only the required variables. This code will create a submit button that will send a charge of $1.00 to a store with a configuration ID of AF4Fs1024.

Sample Code - Basic Hosted Payment Page Call with Required Variables

```
<FORM METHOD=”POST” ACTION=https://esqa.moneris.com/HPPDP/index.php>
<INPUT TYPE=”HIDDEN” NAME=”ps_store_id” VALUE=”AF4Fs1024”>
<INPUT TYPE=”HIDDEN” NAME=”hpp_key” VALUE=”Hsjh4GSr4g”>
<INPUT TYPE=”HIDDEN” NAME=”charge_total” VALUE=”1.00”>
<!--MORE OPTIONAL VARIABLES CAN BE DEFINED HERE -->
<INPUT TYPE=”SUBMIT” NAME=”SUBMIT” VALUE=”Click to proceed to Secure Page”>
</FORM>
```

Optional Variables for Basic Transaction

Item Details can be sent so that they may be presented in the receipt or email that Moneris Gateway generates. Moneris Gateway will also store the item data so that it may be reviewed through the Moneris GatewayMerchant Resource Center.Moneris Gateway does not perform any calculations to arrive at a final amount.

Things to Know

- Where ***n*** is an alphanumeric value less than 10 characters long, unique to each item
- You must send a quantity***n*** > 0 or the item will not be added to the item list
- For each item all five variables should be included
- Items will be stored in the Merchant Resource Centre and
  - will be included in the email receipt if **Include Line Item Details** is selected in the Email Receipt Configuration
  - will be displayed on the payment page if **Display Items Details** is selected in the Appearance Configuration

Table 1: Optional Variables - Item Details

Variable name	Type	Description
idn	hidden	Product Code - SKU (max 10 chars)
descriptionn	hidden	Product Description - (max 15 chars)
quantityn	hidden	Quantity of Goods Purchased - (max - 4 digits)
pricen	hidden	Unit Price - (max - "7"."2" digits, i.e. min 0.00 & max 9999999.99)
subtotal***n***	hidden	Quantity X Price of Product - (max - "7"."2" digits, i.e. min 0.00 & max 9999999.99)

NOTE: You must send a quantityn or the item will not be added to the item list.

Sample with Item Details

The code below will create one item and define its details. It will set the product id***n***, descriptionn, quantityn,pricen and subtotal***n*** details for the item to be displayed in the receipt.

Sample - Basic Transaction with Item Details

```
<input type="hidden" name="id1" value="1">
<input type="hidden" name="description1" value="Blue Suede Shoes">
<input type="hidden" name="quantity1" value="3">
<input type="hidden" name="price1" value="40.00">
<input type="hidden" name="subtotal1" value="120.00">
```

Other Details for Basic Transaction

These miscellaneous fields can be submitted as part of the transaction request.

Table 1: Optional Variables - Other Transaction Variables

Variable name	Type	Description
cust_id	hidden	This is an ID field that can be used to identify the client, commonly used for student #s, policy #s, client name or invoice #s. Can not be more than 50 characters.
order_id	hidden	MUST be unique per transaction and be no more than 50 characters. System will generate a value if excluded.

lang	hidden	This defines what language the Hosted Pay Page and the receipts will be in:
		en-ca = English
		fr-ca = French
		If the tag is not included the Hosted Payment Page will default to English.

gst	hidden	This is where you would include Goods and Services Tax charged, should you wish it to be displayed on the items list. (min 0.00 & max 9999999.99)
pst	hidden	This is where you would include Provincial Sales Tax charged, should you wish it to be displayed on the items list. (min 0.00 & max 9999999.99)

Variable name	Type	Description
hst	hidden	This is where you would include Harmonized Sales Tax charged, should you wish it to be displayed on the items list. (min 0.00 & max 9999999.99)
shipping_cost	hidden	This is where you would include shipping charges, should you wish it to be displayed on the items list. (min 0.00 & max 9999999.99)
note	text	This is any special instructions that you or the cardholder might like to store. Can not be more than 50 chars.
email	text	Customer's email address. This address will be used for the email receipts. Cannot be more than 50 characters. If you have chosen to send an email receipt to the cardholder this field must be included.
eci	hidden	If using the Hosted Payment Page to integrate an internal order management system for Mail/Telephone Orders, send an ecivalue of 1. If the Hosted Pay Page supports VbV/MCSC, the eci generated by VbV/MCSC will override the value passed in.

NOTE:

The order_id variable allows the following characters: a-z A-Z 0-9 _ - : . @ spaces

All other request fields allow the following characters: a-z A-Z 0-9 _ - : . @ $ = /

Sample with Other Transaction Details

The code below will set the cust_id, order_id, and the lang as well as add gst, pst, hst and shipping cost details for display in the receipt.

Sample - Basic Transaction with Other Transaction Details

```
<INPUT TYPE="HIDDEN" NAME="cust_id" VALUE="invoice: 123456-12-1">
<INPUT TYPE="HIDDEN" NAME="order_id" VALUE="oid43333">
<INPUT TYPE="HIDDEN" NAME="lang" VALUE="fr-ca">
<INPUT TYPE="HIDDEN" NAME="gst" VALUE="0.80">
<INPUT TYPE="HIDDEN" NAME="pst" VALUE="0.70">
<INPUT TYPE="HIDDEN" NAME="hst" VALUE="1.50">
<INPUT TYPE="HIDDEN" NAME="shipping_cost" VALUE="4.99">
<INPUT TYPE="HIDDEN" NAME="eci" VALUE="1">
```

Shipping and Billing Information

Shipping and billing information will be stored in the Merchant Resource Center. It may also be included in the email receipt if **Include billing details** and/or **Include shipping details** are selected in the Email Receipt Configuration.

These fields may also be returned in the transaction response or displayed on the response receipt if **Return billing details** and/or **Return shipping details** are selected in the Response/Receipt Field Configuration.

NOTE: Each of the fields in the table below is alphanumeric and cannot be more than 30 characters.

Table 1: Shipping and Billing Address Variables

Variable name	Type
ship_first_name	text
ship_last_name	text
ship_company_name	text
ship_address_one	text
ship_city	text
ship_state_or_province	text
ship_postal_code	text
ship_country	text
ship_phone	text
ship_fax	text
bill_first_name	text
bill_last_name	text
bill_company_name	text
bill_address_one	text
bill_city	text

Variable name	Type
bill_state_or_province	text
bill_postal_code	text
bill_country	text
bill_phone	text
bill_fax	text

The code below will set the billing and shipping details so they may be stored in the Merchant Resource Center. Also, depending on the Hosted Payment Page configuration, these fields may also be included in the response and receipt.

Sample Set Billing and Shipping Details - CA

```
<INPUT TYPE="HIDDEN" NAME="bill_first_name" VALUE="John">
<INPUT TYPE="HIDDEN" NAME="bill_last_name" VALUE="Smith">
<INPUT TYPE="HIDDEN" NAME="bill_company_name" VALUE="Moneris Solutions">
<INPUT TYPE="HIDDEN" NAME="bill_address_one" VALUE="101 Main St">
<INPUT TYPE="HIDDEN" NAME="bill_city" VALUE="Smallville">
<INPUT TYPE="HIDDEN" NAME="bill_state_or_province" VALUE="NT">
<INPUT TYPE="HIDDEN" NAME="bill_postal_code" VALUE="Z1Z 1Z1">
<INPUT TYPE="HIDDEN" NAME="bill_country" VALUE="Canada">
<INPUT TYPE="HIDDEN" NAME="bill_phone" VALUE="555-555-5555">
<INPUT TYPE="HIDDENV NAME="bill_fax" VALUE="555-555-6666">
<INPUT TYPE="HIDDEN" NAME="ship_first_name" VALUE="Jen">
<INPUT TYPE="HIDDEN" NAME="ship_last_name" VALUE="Smith">
<INPUT TYPE="HIDDEN" NAME="ship_company_name" VALUE="Moneris Solutions">
<INPUT TYPE="HIDDEN" NAME="ship_address_one" VALUE="150 Lakeshore Rd">
<INPUT TYPE="HIDDEN" NAME="ship_city" VALUE="Springfield">
<INPUT TYPE="HIDDEN" NAME="ship_state_or_province" VALUE="IL">
<INPUT TYPE="HIDDEN" NAME="ship_postal_code" VALUE="234567">
<INPUT TYPE="HIDDEN" NAME="ship_country" VALUE="USA">
<INPUT TYPE="HIDDEN" NAME="ship_phone" VALUE="333-555-5555">
```

NOTE:

If you send characters that are not included in the allowed list, the extra transaction details (e.g., bill_ or ship_) may not be stored, included in the receipt or in the response. The following are allowed characters:

a-z A-Z 0-9 _ - : . @ $ = /

Also, the data sent in Billing and Shipping Address variables will not be used for any address verification.

Please refer to "Address Verification Service (AVS)" (page 86) for information on Address Verification Service (AVS).

Optional 'rvar' Variables

Things to Know:

- Where ***n*** is an alphanumeric value less than 10 characters long, unique to each rvar variable.
- The data sent in the rvar variables will NOT be stored in the Merchant Resource Center. These fields will be echoed backin the transaction response – in a GET or POST method.
- They may also be sent in the email receipt to the merchant if **Include 'rvar' in merchant email** is selected in the Email Receipt Configuration.

Table 1: Optional Variables - rvar

Variable name	Type	Description
rvar***n***	hidden	If these extra variables are sent in the request, they will be echoed back in the response (if GET or POST have been selected for the Response Method). Commonly used for session IDs. These variables must begin with “rvar” and then contain any alphanumeric string (i.e. rvar1, rvarname, rvarMyVariable).

Sample Optional 'rvar' Request

```
<INPUT TYPE="HIDDEN" NAME="rvar1" VALUE="1">
<INPUT TYPE="HIDDEN" NAME="rvar_monkey" VALUE="monkeys are funny">
<INPUT TYPE="HIDDEN" NAME="rvar_123abc" VALUE="123abc">
```

3.1.2.2 INTERAC® Online Payment Transaction

No alteration needs to be made to the transaction request.

When Moneris Gatewayreceives the transaction request the Hosted Payment Page will automatically determine whether INTERAC® Online Payment is enabled for your store and create a Payment Option screen.

Note that INTERAC® Online Payment only supports the Purchase transaction. If your default transaction is PreAuth all credit card transactions will be processed as a PreAuth and will need to be subsequently captured,INTERAC® Online Payment transactions will be processed as a Purchase and do not require any further action.

3.1.2.3 Performing a Transaction Risk Management Tool Transaction

In order to perform a Transaction Risk Management Tool transaction it is essential that you configure the Hosted Paypage accordingly. If the Hosted Paypage is properly configured, the Hosted Payment Page will automatically generate a unique session_id and send the request to ThreatMetrix for device profiling. You will then receive several variables in the GET or POST response. If Moneris Gateway generates the receipt on the merchant’s behalf, the Transaction Risk Management Tool details will not be displayed.

The results will be visible in the Merchant Resource Center. The Transaction Risk Management Tool can only be performed once on a given transaction.

Once Moneris Gateway receives the device profiling response from ThreatMetrix, a response is then returned with the Transaction Risk Management Tool transaction and a risk score. This response is sent in the format, and to the URL, defined in the "Response Method" portion of the Hosted Payment Page configuration. Please see the table below for a list of possible Transaction Risk Management Tool responses.

For a list of Rule Names and Rule Codes please refer to Transaction Risk Management Tool Rules & Codes (see page 93).

In addition to the Transaction Risk Management Tool, an Assertion Query transaction can be performed if merchant investigates a transaction and determines that it is fraudulent. An Assertion Query can be done manually through the Moneris Gateway Transaction Risk Management Tool API. If you are interested in also being able to send an Assertion Query via your online application, please refer to the Moneris Gateway Transaction Risk Management Tool API found on our download portal at https://github.com/moneris/.

3.1.3 How Do I Initiate a Recurring Billing Transaction?

Moneris Gateway offers an optional feature to process your regular recurring payments. This is often used for subscriptions, memberships or any time a fixed amount is charged at a regular interval. The transaction is sent to Moneris Gateway using the same methods listed above and the fields listed below are added to the POST. Recurring Billing must be enabled on your merchant account — please contact Moneris sales to have this feature added to your profile if you have not already done so.

Table 1: Recurring Billing - mandatory variables

Variable name	Type	Description
doRecur	hidden	1 = initiate a Recurring transaction, anything else will not initiate a recurring transaction
recurUnit	hidden	Must be "day", "week" "month" or "eom" (end of month). This is the base unit for the recurring interval.
recurPeriod	hidden	Numeric value. The period is used in conjunction with recurUnit to determine the interval between payments. **EXAMPLE:** recurUnit = "month" and recurPeriod = "12" the

Variable name	Type	Description
		charge will be billed once every twelve months. If recurUnit= "week" and recurPeriod = "1" the charge will be billed once a week. If recurUnit = "eom" and recurPeriod = "3" the charge will be billed every 3 months (on the last day of the month). Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future.
recurStartDate	hidden	Must be in the format "YYYY/MM/DD" – this determines the date of the first charge. This date must be in the future; it cannot be the date the transaction is sent (Please see recurStartNow to bill card holder immediately).
recurStartNow	hidden	"true" / "false" This will charge the transaction immediately and then initiate recurring billing to commence on the recurStartDate. **EXAMPLE:** To charge a card immediately (assuming the date is May 9, 2006) and then bill every month thereafter: recurStartNow = "true" recurStartDate = "2006/05/09" – set the date 1 month after the present recurUnit = "month"

Variable name	Type	Description
		recurPeriod = "1"

recurAmount	hidden	Amount to charge on a recurring basis - no $, must include 2 decimal places (i.e. 3.04). This can vary from the charge_total. If using recurStartNow, charge_total is used for the immediate transaction and recurAmount is used for every transaction in the future.
		EXAMPLE: A member is joining halfway through May 2006 – you would like to bill the remaining half of the month ($20.00) and then bill them on the first day of the month every month for the full month ($40.00) starting June 1st 2006.
		charge_total = "20.00"
		recurAmount = "40.00"
		recurStartNow = "true"
		recurStartDate = "2006/06/01"
		recurUnit = "month"
		recurPeriod =”1”

recurNum	hidden	The number of times to process the recurring charge. **EXAMPLE:** if you are billing a client monthly for one year recurNum= "12" We advise against setting a period of longer than 5 years. The suggested maximum should be calculated using your

Variable name	Type	Description
		recurUnit and recurPeriod settings over a 5 year period. **EXAMPLE:** If you would like to bill a client indefinitely you should set the recurNum to be approximately 5 years in the future – so if you are billing once a month recurNum = "60" (12 x 5) or if you are billing every two weeks recurNum = "130" (26 x 5)

NOTE: When completing the recurring billing portion please keep in mind that to prevent the shifting of recur bill dates, avoid setting the start_date for anything past the 28th of any given month when using the recur_unit set to “month”. For example, all billing dates set for the 31st of May will shift and bill on the 30th in June and will then bill the cardholder on the 30th for every subsequent month. To set the billing dates for the end of the month please set the recur_unit to “eom”.

Below is a sample of the HTML to initiate a recurring transaction – this must be sent as part of a Hosted Payment Page request that includes: hpp_key, ps_store_id and charge_total.

This transaction will bill the amount sent in charge_total immediately (using the example above, charge_total=”1.00”) Then commencing June 1, 2006 (assuming that this is a date in the future) the card will be billed $4.00 every 2 weeks, 26 times (one year).

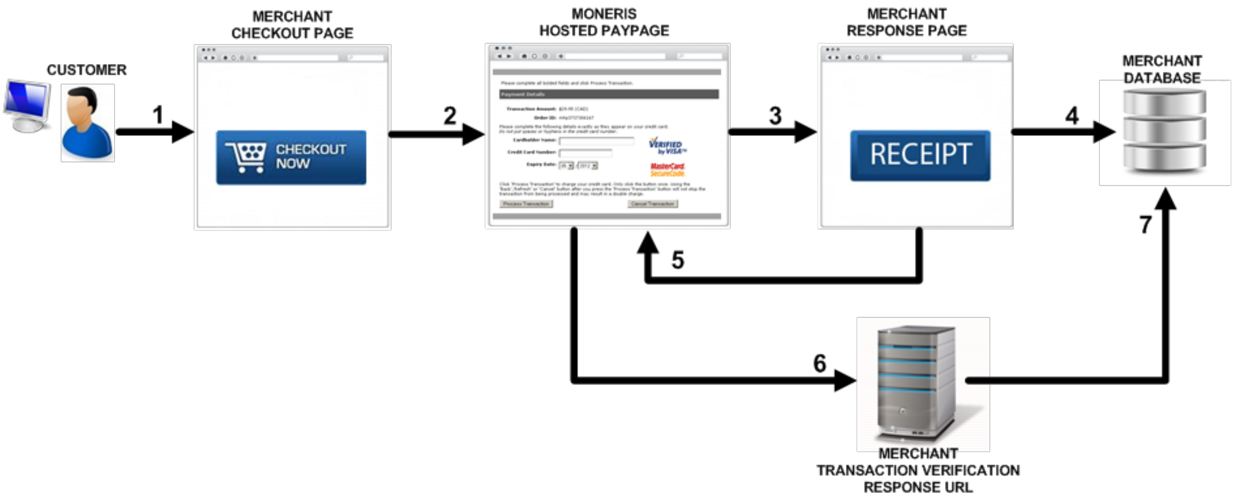
Sample Initiate Recurring Transaction

```
<input type="hidden" name=doRecur value ="1">
<input type="hidden" name=recurUnit value ="week">
<input type="hidden" name=recurStartDate value="2006/06/01">
<input type="hidden" name=recurNum value="26">
<input type="hidden" name=recurStartNow value ='true'>
<input type="hidden" name=recurPeriod value='2'>
<input type="hidden" name=recurAmount value='4.00'>
```

3.1.4 Transaction Verification with Hosted Payment Page

- 3.1.1 Hosted Payment Page Process Flow
- 3.1.4.2 Performing a Transaction Verification

3.1.4.1 Hosted Payment Page Process Flow - Transaction Verification



Outlined above is the Hosted Payment Page flow with the Transaction Verification feature enabled.

The steps required to achieve this are as follows:

1. The customer arrives on the merchant's website (application). At this point the merchant must determine the amount of the transaction and also collect any additional data.
2. Once the cardholder is ready to pay, the merchant's checkout page will submit an HTTP form POST to the Hosted Payment Page . At this time, the customer will be redirected from the merchant's website to the Moneris Hosted Payment Page URL. For details on the HTTP form POST, please refer to Performing a Transaction Verification (see page 44) which outlines the mandatory and optional fields that may be sent to the Hosted Payment Page. On the Hosted Payment Page, the cardholder will fill in their secure payment details such as their card number and submit the transaction. At this time, Moneris will process the transaction and build a response.
3. Moneris will redirect the customer back to the merchant's website as well as send the response back to the merchant in a POST or GET format to the Response URL provided in the Hosted Payment Page Configuration as described in Hosted Payment Page Configuration Tool (see page 12).
4. Once the merchant receives the response details they may provide a receipt to the customer at this time and then may store these details for future purposes such as reporting or tracking.
5. Once of the fields returned in the response (step #3 above) is the transactionKey. The merchant may now send this transactionKey back to the Moneris Hosted Payment Page using another HTTP form POST. For details on the transaction verification POST, please refer to section Performing a Transaction Verification (see page 44).
6. Once the Moneris Hosted Payment Page receives this transactionKey it will look up the details we have on file for this transaction and send the amount, response code and message back to the merchant in one of the following formats: POST, GET, key/value pairs, or display XML on our server. This response will be sent to the Transaction Verification Response URL provided in the Hosted Payment Page Configuration as described in Hosted Payment Page Configuration Tool

(see page 12). NOTE: the merchant may set the Response URL (step 3) and Transaction Verification URL (step 6) to be one and the same or set 2 different URLs.

7. Once the merchant receives the additional transaction verification response details, they may now display a receipt to the customer (if this was not yet done in step 4 above) and also compare this response to the one received in step 3 above to verify the accuracy of the original response data.

3.1.4.2 Performing a Transaction Verification

In order to perform a Transaction Verification it is essential that you configure the Hosted Payment Page accordingly. If the Hosted Payment Page is properly configured you will receive a variable in the GET or POST response called “transactionKey”.

It is advised that you log the initial transaction response and then compare the Transaction Verification response to ensure authenticity. You can use a redirect (for example, onLoad=”submit”) or any other method to submit the request. We suggest automating the Transaction Verification and not using a button to submit the information.

Transaction Verification can only be performed once on a given transaction, and it can only be performed within 15 minutes of the original transaction.

Table 1: Variables for Transaction Verification - Hosted Payment Page

Variable name	Type	Description
	form	Development and testing: https://esqa.moneris.com/HPPDP/index.php Production: https://www3.moneris.com/HPPDP/index.php
ps_store_id	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool
hpp_key	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool
transactionKey	hidden	This is returned in the transaction response - refer to Adding a new profile to Vault using the Hosted Vault (see page 62).

Below is a sample of the Transaction Verification Request:

Sample Transaction Verification Request

```
<FORM NAME=VERIFY ACTION="https://esqa.moneris.com/HPPDP/verifyTxn.php">
<INPUT TYPE="HIDDEN" NAME="ps_store_id" VALUE="qampg">
<INPUT TYPE="HIDDEN" NAME="hpp_key" VALUE="hpPu7yr4Hn5k">
<INPUT TYPE="HIDDEN" NAME="transactionKey" VALUE="SDGJSGI4ujwrlsfGSGsjhlfhs5lsffs4h">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to perform verification">
</FORM>
```

Once Moneris Gateway receives the transaction verification request it decrypts the key, verifies and logs the request. A response is then returned with the transaction information and a status. This response is sent in the format, and to the URL, defined in the "Security Features" portion of the Hosted Payment Page configuration. Please see the table below for a list of possible Transaction Verification statuses.

Table 2: Response Fields - Transaction Verification Requests

Variable name	Type	Description
response_order_id	50-character alphanumeric	order_id of the original transaction
response_code	3-character alphanumeric	Transaction Response Code from the original transaction < 50: Transaction approved >= 50: Transaction declined NULL: Transaction was not sent for authorization * If you would like further details on the response codes that are returned please see the Response Codes document available for download at: https://developer.moneris.com
amount	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	This is the amount of the original transaction (i.e. min 0.01 & max 9999999.99)
txn_num	20-character alphanumeric	Gateway Transaction identifier from the original transaction.

Variable name	Type	Description
transactionKey	100-character alphanumeric	The transactionKey from the request
status	alphanumeric	This is the value to check to see if the transaction has been properly validated. Below is a list of possible replies and their meaning. Valid-Approved: The transaction was approved and successfully validated Valid-Declined: The transaction was declined and successfully validated Invalid: No reference to this transactionKey, validation failed Invalid-ReConfirmed: An attempt has already been made with this transaction key, validation failed Invalid-Bad_Source: The Referring URL is not correct, validation failed

Table 3: Error Codes - Transaction Verification

Code	Message/Description
991	Invalid referrer URL - <referrer url>: Referring URL does not match what is listed in the "Security Features" portion of the Hosted Payment Page configuration, validation failed. The source URL will be returned.
994	Invalid – Reconfirmed: The transaction has already been confirmed, validation failed.
995	Invalid: Not a valid confirmation request. Either the transaction doesn't exist or the request is older than 15 minutes, validation failed.

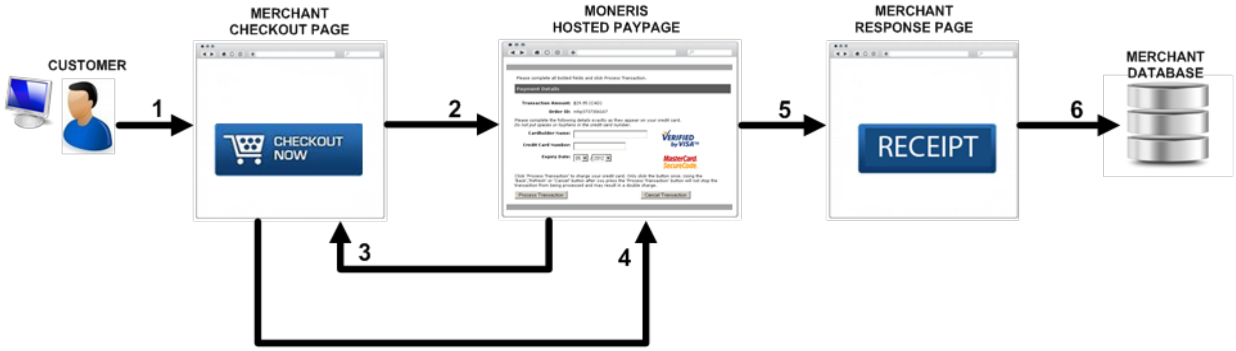
Below is a sample of a valid Transaction Verification Response displayed on our server in XML format:

Sample Transaction Verification Response (Valid)	Sample Transaction Verification Response (Invalid)
<?xml version="1.0" standalone="yes"?> <response> <order_id>ord-140708-1705</order_id> <response_code>27</response_code> <amount>15.00</amount> <txn_num>381038-0_7</txn_num> <status>Valid-Approved</status> <transactionKey>agtA20RwCQ60y3jA49KlaxTMYSQdS4 </transactionKey> </response>	<?xml version="1.0" standalone="yes"?> <response> <order_id>ord-140708-1705</order_id> <response_code>994</response_code> <amount>null</amount> <txn_num>381038-0_7</txn_num> <status>Invalid-ReConfirmed</status> <transactionKey>agtA20RwCQ60y3jA49KlaxTMYSQdS4 </transactionKey> </response>

3.1.5 Data Preload with Hosted Payment Page

The Hosted Payment Page can also be configured to use data preload functionality. We highly recommend that you enable this feature as it enhances the security of your Hosted Payment Page.

3.1.5.1 Hosted Payment Page Process Flow - Data Preload



Outlined above is the Hosted Payment Page flow with the Data Preload feature implemented.

The steps required to achieve this are as follows:

1. The customer arrives on the merchant's website (application). At this point the merchant must determine the amount of the transaction and also collect any additional data.
2. Once the cardholder is ready to pay, the merchant's checkout page will submit an HTTPS POST using a server side programming language to the Moneris Hosted Payment Page sending over all of the transaction data. NOTE: the cardholder will not be redirected yet.
3. The Hosted Payment Page will store this transaction data and respond back to the merchant's site by sending an XML response containing a ticket number.

4. The merchant's checkout site will need to collect this response data, and build a new form POST to the Moneris Hosted Payment Page. At this time, the customer will be redirected from the merchant's website to the Moneris Hosted Payment Page URL.
5. On the Hosted Payment Page, the cardholder will fill in their secure payment details such as their card number and submit the transaction. At this time, Moneris will process the transaction and build a response.
6. Once the merchant receives the response details they must provide a receipt to the customer and then may store these details for future purposes such as reporting or tracking.

For details on the Preload form POST data, please refer to Implementing Enhanced Hosted Payment Page- Data Preload (see page 48) which outlines the mandatory fields.

3.1.5.2 Implementing Enhanced Hosted Payment Page- Data Preload

- 3.1.5.1 Hosted Payment Page Process Flow - Data Preload
- Step 1 - Preload Data Request
- Step 2 - Handling the Preload Response
- Step 3 - Proceed to Hosted Payment Page
- Step 4 - Normal Hosted Payment Page Process

This feature allows merchants to preload transaction data into the Hosted Pay Page through a direct server to server request. The Hosted Payment Page then returns an XML response containing a "ticket". The ticket is then sent with the ps_store_id through the shopper's browser in a request to theHosted Payment Page. The Hosted Payment Page will then link the preloaded data to the browser request by using the ticket information after which the process will proceed as a regular Hosted Payment Page transaction. In a typical shopping cart checkout experience you must manage and ensure that preload requests correspond to the correct shopper's browser session.

Overview of the Preload Process

1. Preload Data Request
2. Handling the Preload Response
3. Proceed to Hosted Payment Page
4. Normal Hosted Payment Page Process

Step 1 - Preload Data Request

The fields below need to be sent via an HTTPS POST using a server side programming language such as .NET, Java, and PHP. Other optional variables could also be passed in this step, such as order ID, customer ID, billing and shipping information, etc.

Table 1: Data Preload Request Fields

Variable name	Description
URL's to POST to	https://esqa.moneris.com/HPPDP/index.php - Development https://www3.moneris.com/HPPDP/index.php - Production
ps_store_id	Provided by Moneris Solutions – Hosted Pay Page Configuration Tool
hpp_key	Provided by Moneris Solutions – Hosted Pay Page Configuration Tool
hpp_preload	Blank value. This is used to indicate that the transaction is a preload transaction
charge_total	Amount to charge, must be have 2 decimal places
order_id	(Optional) Merchant defined unique transaction identifier - must be unique for every transaction attempt. Characters allowed for Order ID: a-z A-Z 0-9 _ - : . @ spaces **NOTE:** only the last 10 characters of the Order ID will appear in the Moneris Merchant Direct report

Step 2 - Handling the Preload Response

After you send the preload data request to Moneris Gateway you will receive an XML response with:

- the hpp_id
- the ticket
- the order_no, and
- the response code.

The combination of the hpp_id, ticket, and the order ID will uniquely identify this particular set of preloaded data with what's stored already on the Moneris side. To make this work you must ensure the shopper's browser session is linked to the correct ticket.

Table 1: Data Preload Response Fields

Variable name	Description
hpp_id	The ps_store_id is returned as the hpp_id in the XML response.
ticket	The unique value assigned to the preload transaction
order_id	The order_id that was sent in the preload request; if no order id is sent in the preload request then a unique Order ID will be assigned by the Moneris system.
response_code	Transaction Response Code < 50: data successfully loaded >= 50: data not loaded

Below is a sample of a valid data preload Response displayed on our server in XML format

Sample XML Response for Data Preload

```
<?xml version="1.0" standalone="yes"?>
<response>
<hpp_id>4YAHJqa002</hpp_id>
<ticket>hppEzskxQTZe8DbO80ga</ticket>
<order_id>hpp_test_1</order_id>
<response_code>1</response_code>
</response>
```

Step 3 - Proceed to Hosted Payment Page

Upon receiving confirmation from the user that they are ready to proceed, you would then redirect the shopper via an HTTPS post with the variables below to the Moneris Hosted Payment Page.

Table 1: Proceed to Hosted Payment Page Fields

Variable name	Description
URL's to POST to	https://esqa.moneris.com/HPPDP/index.php - Development https://www3.moneris.com/HPPDP/index.php - Production
hpp_id	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool
hpp_preload	Blank value. This is used to indicate that the transaction is a preload transaction
ticket	A value returned by the preload request which helps identify the transaction

NOTE: Other optional variables should not be sent, **only** the above variables should be sent in this request.

Sample Proceed to Hosted Payment Page

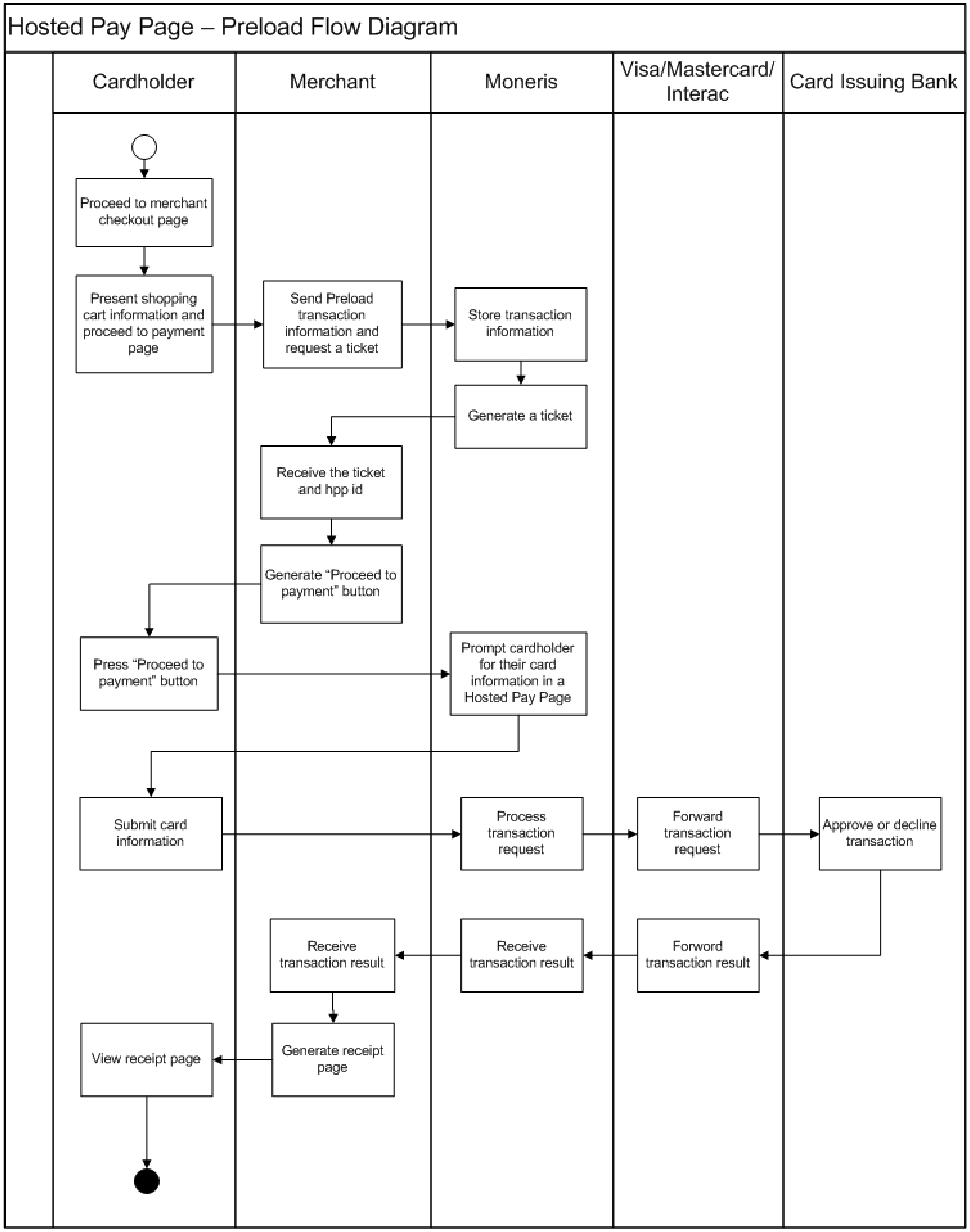
```
<FORM METHOD="POST" ACTION="https://esqa.moneris.com/HPPDP/index.php">
<INPUT TYPE="HIDDEN" NAME="hpp_id" VALUE="4YAHJqa002">
<INPUT TYPE="hidden" NAME="hpp_preload" >
<INPUT TYPE="hidden" NAME="ticket" VALUE="hppEzskxQTZe8DbO80ga">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

Step 4 - Normal Hosted Payment Page Process

At this point the user will reach the Hosted Payment Page where they will enter their credit card information on a secure Moneris website. After the user completes their payment they can then be sent to a merchant generated receipt page and the result of the transaction can be recorded in the merchants order management system (if applicable).

If the cardholder does not proceed with charging the credit card at this point, then the ticket will become invalid. A new preload request will have to be created if the cardholder wishes to checkout.

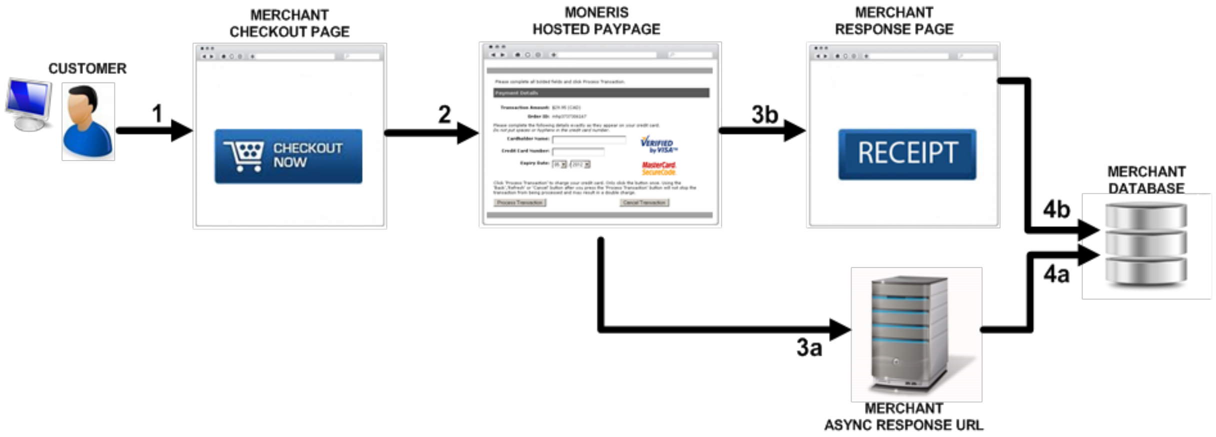
Data Preload Diagram


Hosted Pay Page – Preload Flow Diagram
Cardholder
Merchant
Moneris
Visa/Mastercard/ Interac
Card Issuing Bank
Proceed to merchant checkout page
Present shopping cart information and proceed to payment page
Send Preload transaction information and request a ticket
Store transaction information
Generate a ticket
Receive the ticket and hpp id
Generate "Proceed to payment" button
Press "Proceed to payment" button
Prompt cardholder for their card information in a Hosted Pay Page
Submit card information
Process transaction request
Forward transaction request
Approve or decline transaction
Forword transaction result
Receive transaction result
Receive transaction result
Generate receipt page
View receipt page

3.1.6 Asynchronous Transactions in Hosted Payment Page

The Hosted Payment Page can be configured for asynchronous transaction functionality. We highly recommend that you enable this feature as it enhances the security of your Hosted Payment Page.

3.1.6.1 Hosted Payment Page Process Flow with Asynchronous Transaction Response



Outlined above is the Hosted Payment Page flow with the Asynchronous Transaction Response feature enabled.

The steps required to achieve this are as follows:

1. The customer arrives on the merchant’s website (application). At this point the merchant must determine the amount of the transaction and also collect any additional data.
2. Once the cardholder is ready to pay, the merchant’s checkout page will submit an HTTP form POST to the Moneris Hosted Payment Page (HPP). At this time, the customer will be redirected from the merchant’s website to the Moneris Hosted Pay Page URL. For details on the HTTP form POST, please refer to Hosted Payment Page Configuration Tool (see page 12), which outlines the mandatory and optional fields that may be sent to the Hosted Payment Page. On the Hosted Payment Page, the cardholder will fill in their secure payment details such as their card number and submit the transaction. At this time, Moneris Gateway will process the transaction and build a response.
3. 2 responses will be sent out
   a. Moneris Gateway will perform a server to server POST of the response data to the Async Response URL provided in the Hosted Payment Page Configuration as described in Hosted Payment Page Configuration Tool (see page 12).
   b. Moneris Gateway will also send an additional response back to the merchant in a POST or GET format to the Response URL provided in the Hosted Payment Page Configuration as described in 2.2.1 Configuring the Hosted Payment Page on page 15 of this document. The customer will also be redirected back to the merchant’s website to this same Response URL.

4.

   a. Once the merchant receives the Asynch response details they may store these details for future purposes, such as reporting, as well as use it to compare against the response received in step 3B above.
   b. Once the merchant receives the response details they must provide a receipt to the customer and then may store these details for future purposes such as reporting or tracking. They may also at this time compare this response to the one received in step 3A above to verify the accuracy of the data.

3.1.6.2 Implementing the Asynchronous Transaction Response

The Asynchronous Transaction Response feature will perform a server to server POST of the response data as a secondary method of getting the response data. This does not replace the normal transaction response which will still be sent through the browser as a POST or a GET. This is a supplementary feature that can be used to verify/validate the browser response.

If this feature is enabled in the Hosted Payment Page Configuration this POST will automatically be sent back to the Async Response URL once the transaction processing is complete.

Once the merchant receives this response data, it may be used to compare to and verify the original response.

The following is a sample of an Asynchronous Transaction Response:

Sample Asynchronous Transaction Response - XML

```
xml_response=
<response>
<response_order_id>mhp13150144643p28</response_order_id>
<bank_transaction_id>660035510010258730</bank_transaction_id>
<response_code>025</response_code>
<iso_code>00</iso_code>
<bank_approval_code>784810</bank_approval_code>
<time_stamp>14:47:45</time_stamp>
<date_stamp>2013-05-31</date_stamp>
<trans_name>purchase</trans_name>
<message>APPROVED * APPROVAL =</message>
<charge_total>1.00</charge_total>
<cardholder>test</cardholder>
<card_num>3735***5005</card_num>
<card>AX</card>
<expiry_date></expiry_date>
<result>1</result>
</response>
```

The following is a sample PHP script used to read the Asynch Transaction Response:

Sample Script to Read Asynchronous Transaction Response

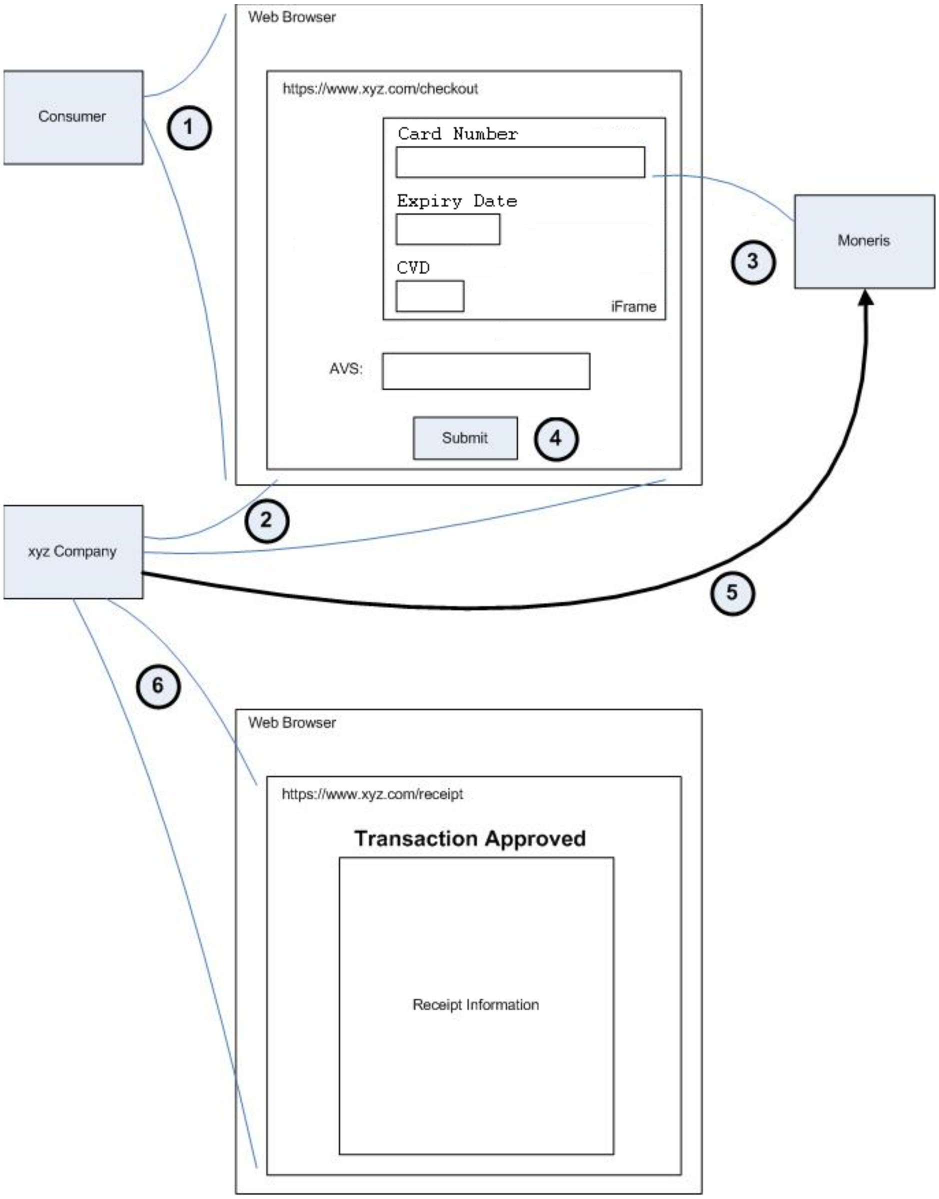
```
<?php
// Recieve the response from the Moneris server
$response = $_REQUEST["xml_response"];
/* remove <?xml version=\'1.0\' standalone=\'yes\'?>
from the string of XML before trying to parse the XML */
$position = strpos($response, "?>");
$length = strlen($response);
$response = substr ($response, $position+2, $length );
// create an array of results
$xml = simplexml_load_string($response);
foreach($xml->children() as $child)
{
$receipt[$child->getName()] = $child;
}
// Prepare to write the results to the log file
$timestamp = date("H:i:s d/m/y", time());
$results = $timestamp . "\r\n";
$results .= "Response Order ID: " . $receipt["response_order_id"] . "\r\n";
$results .= "Bank Transaction ID: " . $receipt["bank_transaction_id"] . "\r\n";
$results .= "Response Code: " . $receipt["response_code"] . "\r\n";
$results .= "ISO Code: " . $receipt["iso_code"] . "\r\n";
$results .= "Bank Approval Code: " . $receipt["bank_approval_code"] . "\r\n";
$results .= "Time Stamp: " . $receipt["time_stamp"] . "\r\n";
$results .= "Date Stamp: " . $receipt["date_stamp"] . "\r\n";
$results .= "Transaction Name: " . $receipt["trans_name"] . "\r\n";
$results .= "Message: " . $receipt["message"] . "\r\n";
$results .= "Charge Total: " . $receipt["charge_total"] . "\r\n";
$results .= "Cardholder Name: " . $receipt["cardholder"] . "\r\n";
$results .= "Card Number: " . $receipt["card_num"] . "\r\n";
$results .= "Card Type: " . $receipt["card"] . "\r\n";
$results .= "Expiry Date(YYMM): " . $receipt["expiry_date"] . "\r\n";
$results .= "Result: " . $receipt["result"] . "\r\n";
//write the results to the log file
$logFile = "async_log.txt";
$fh = fopen($logFile, 'a') or die("can't open file");
fwrite($fh, $results);
fclose($fh);
```

For further information about these response details, please refer to Hosted Payment Page Configuration Tool (see page 12).

3.2 Developing for Hosted Tokenization

- 3.2.1 Hosted Tokenization Process Flow
- 3.2.2 Sending a Hosted Tokenization Request

3.2.1 Hosted Tokenization Process Flow



1. The cardholder shops at a merchant site with their web-browser and ready to check out.
2. The check-out page is presented by the merchant's server with Hosted Tokenization integration.
3. A small portion of the merchant's check-out page has an iframe that links to Moneris' Hosted Tokenization configuration. The text-boxes to collect the credit card data are presented by Moneris.
4. The cardholder enters the credit card data and other payment-related information that the merchant may need in order to process a financial transaction to charge the card. Once the cardholder presses the Submit button, the initial code behind the check-out page submits a request to the Moneris Moneris Gateway to obtain the temporary token that represents the credit card data. The latter code behind the check-out page then takes the token and other

payment-related information from the check-out page and submits them to the merchant's choice of URL that collects the submitted information.

5. The merchant's server sends a Vault transaction to Moneris using the payment information collected by the URL in step 4. Information in the response to the Vault transaction is saved for reference. For more information on Vault, please refer to our Developer Portal at https://developer.moneris.com/
6. Result of the financial transaction is displayed to the cardholder.

3.2.2 Sending a Hosted Tokenization Request

- 3.2.2.1 Getting a Temporary Token
- 3.2.2.2 Forwarding a Temporary Token to Payment Processing Page
- 3.2.2.3 Processing the Payment - Hosted Tokenization

3.2.2.1 Getting a Temporary Token

To get a temporary token you will need to send a request to Moneris from within an iframe. A sample code is illustrated below. Note that the Profile ID in the HTML link below will need to be replaced with your own Profile ID from Creating a Hosted Tokenization Configuration (see page 13).

EXAMPLE: If your Profile ID is htCCLFFI2H31LBK then replace the highlighted ID in the sample below with your ID:

NOTE: If you are integrating your hosted payment solution with Internet Explorer 7, refer to Appendix F Internet Explorer 7 Compatibility on page 113

Sample Getting a Temporary Token

```
src="https://esqa.moneris.com/HPPtoken/index.php?id=ht4RXXBKV9T52A8&css_
    body=background:green;&css_textbox=border-width:2px;&css_textbox_pan=width:140px;&enable_
    exp=1&css_textbox_exp=width:40px;&enable_cvd=1&css_textbox_cvd=width:40px"
<html>
<head>
<title> Outer Frame - Merchant Page</title>
<script>
function doMonerisSubmit()
{
var monFrameRef = document.getElementById('monerisFrame').contentWindow;
monFrameRef.postMessage('','https://esqa.moneris.com/HPPtoken/index.php');
return false;
}
var respMsg = function(e)
{
var respData = eval("(" + e.data + ")");
document.getElementById("monerisResponse").innerHTML = e.origin + " SENT " + " - " +
respData.responseCode + "-" + respData.dataKey + "-" + respData.errorMessage;
document.getElementById("monerisFrame").style.display = 'none';
```

Sample Getting a Temporary Token

```
}
window.onload = function()
{
if (window.addEventListener)
{
window.addEventListener ("message", respMsg, false);
}
else
{
if (window.attachEvent)
{
window.attachEvent("onmessage", respMsg);
}
}
}
</script>
</head>
<body>
<div>This is the outer page</div>
<div id=monerisResponse></div>
<iframe id=monerisFrame src="https://esqa.moneris.com/HPPtoken/index.php?id=ht4RXXBKV9T52A8&css_
    body=background:green;&css_textbox=border-width:2px;&css_textbox_pan=width:140px;&enable_
    exp=1&css_textbox_exp=width:40px;&enable_cvd=1&css_textbox_cvd=width:40px" frameborder='0'
    width="200px" height="30px"></iframe>

<input type=button onClick=doMonerisSubmit() value="submit iframe">
</body>
</html>
```

The IFrame is opened by passing the below arguments as a query string to the following url:

QA: https://esqa.moneris.com/HPPtoken/index.php

Production: https://www3.moneris.com/HPPtoken/index.php

Variable name	Definition
Id	Required - Provided by the Hosted Tokenization profile configuration tool in the MRC.
css_body	Required - CSS applied to the body. By default margin and padding is set to 0.
css_textbox	Required - CSS applied to all text boxes in general.
css_textbox_pan	Optional - CSS applied to the pan text box specifically.
enable_exp	Optional - Must be set to 1 for expiry date text box to be displayed
css_textbox_exp	Optional - CSS applied to the expiry date text box specifically.
enable_cvd	Optional - Must be set to 1 for CVD text box to be displayed

Variable name	Definition
css_textbox_cvd	Optional - CSS applied to the CVD text box specifically.
display_labels	Optional – 0 for no labels, 1 for traditional labels, 2 for place holder labels.
css_input_label	Optional – CSS for input labels
css_label_pan	Optional – CSS for card number label
css_label_exp	Optional – CSS for expiry date label
css_label_cvd	Optional – CSS for CVD label
pan_label	Optional – text for card number label (default is "Card Number")
exp_label	Optional – text for expiry date label (default is "Expiry Date")
cvd_label	Optional – text for CVD label (default is "CVD")

The response will be returned as JSON. It will contain 4 arguments:

- responseCode - Indication whether the page-loading or card-submission was successful or why it failed. Please note, if expiry text box or CVD text box are enabled, the returned responseCode value will be in the form of a list (e.g. ["944","943"]), since there may be more than one failure. For example, in the case where both the card number entered and expiry date are invalid. If only the card number text box is displayed, the responseCode will be returned in the form of a string.
- errorMessage - Description of failure (This is a very generic description – see "responseCode Definitions" below for specific error code results).
- bin - BIN range of the submitted card number. Provides merchant ability to determine the card type and perform any card-specific processing.
- dataKey - Tokenized card number. This is what is used with the Vault API transaction.

Table 1: Error Codes - Hosted Tokenization

Code	Message/Description
001	approved
940	Invalid profile id (on tokenization request)
941	Error generating token
942	Invalid Profile ID, or source URL
943	Card data is invalid (not numeric, fails mod10, we will remove spaces)

Code	Message/Description
944	Invalid expiration date (mmyy, must be current month or in the future)
945	Invalid CVD data (not 3-4 digits)

3.2.2.2 Forwarding a Temporary Token to Payment Processing Page

To charge the card using the temporary token you will need to send the temporary token to a page on your site that implements the Moneris Vault API.

The token will be received by the below JavaScript function in your code, this is the part of the page where you would put any code that forwards the token to a secondary page for processing.

Sample Forwarding a Temporary Token

```
var respMsg = function(e)
{
var respData = eval("(" + e.data + ")");
document.getElementById("monerisResponse").innerHTML = e.origin + " SENT " + " - " +
    respData.responseCode + "-" + respData.dataKey + "-" + respData.errorMessage;
document.getElementById("monerisFrame").style.display = 'none';
// your token will be in the filed: respData.dataKey
// from this point in the Javascript you can have more code that posts the token to another page
    that will actually process the payment.
}
```

3.2.2.3 Processing the Payment - Hosted Tokenization

To charge the card on your payment processing page you will need to use one of the Moneris Vault API's. The Moneris Vault API's are available in various programming languages. Below is a Java sample code outlining a Purchase transaction that utilizes the temporary token to charge the card.

Sample Processing the Payment - Hosted Tokenization

```
public class TestResPurchaseCCExpDate
{
public static void main(String args[]) throws IOException
{
/********************** Request Variables ****************************/
String host = "esqa.moneris.com";
String store_id = "store1";
String api_token = "yesguy";
/********************** Transaction Variables ************************/
String data_key = "ot-pILqxjIp3BPZgLgN80roTHrAH"; //Temp Token from Hosted Tokenization process.
String order_id; // Application will provide this unique value.
String cust_id = "Hilton_1";
String amount = "1.00";
String crypt_type = "7";
String exp_date = "1209";
/************************ Request Object ******************************/
ResPurchaseCC resPurchaseCC = new ResPurchaseCC(data_key, order_id, amount, crypt_type);
resPurchaseCC.setCustId(cust_id);
```

Sample Processing the Payment - Hosted Tokenization

```
// IMPORTANT: ideally the expiration date should be set within the iframe (This is done by setting
    the optional expiry text box field to enabled). If not enabled, the expiration date will need
    to be set by the setExpdate call.
// resPurchaseCC.setExpdate(exp_date);
// IMPORTANT note: ideally if the CVD feature is used, the value should be set within the iframe
    (This is done by setting the optional CVD text box field to enabled). If not enabled, the CVD
    value can be passed through the setCvdInfo call. In production, if the CVD feature will be
    used it needs to be enabled on the merchant account.
// CvdInfo cvd = new CvdInfo ("1", "789");
// resPurchaseCC.setCvdInfo(cvd);
ResolverHttpsPostRequest mpgReq =
new ResolverHttpsPostRequest(host, store_id, api_token, resPurchaseCC);
/************************ Receipt Object ******************************/
try
{
ResolverReceipt resreceipt = mpgReq.getResolverReceipt();
ResolveData resdata = resreceipt.getResolveData();
System.out.println("DataKey = " + resreceipt.getDataKey());
System.out.println("ReceiptId = " + resreceipt.getReceiptId());
System.out.println("ReferenceNum = " + resreceipt.getReferenceNum());
System.out.println("ResponseCode = " + resreceipt.getResponseCode());
System.out.println("AuthCode = " + resreceipt.getAuthCode());
System.out.println("Message = " + resreceipt.getMessage());
System.out.println("TransDate = " + resreceipt.getTransDate());
System.out.println("TransTime = " + resreceipt.getTransTime());
System.out.println("TransType = " + resreceipt.getTransType());
System.out.println("Complete = " + resreceipt.getComplete());
System.out.println("TransAmount = " + resreceipt.getTransAmount());
System.out.println("CardType = " + resreceipt.getCardType());
System.out.println("TxnNumber = " + resreceipt.getTxnNumber());
System.out.println("TimedOut = " + resreceipt.getTimedOut());
System.out.println("ResSuccess = " + resreceipt.getResSuccess());
System.out.println("PaymentType = " + resreceipt.getPaymentType() + "\n");
//Contents of ResolveData
System.out.println("Cust ID = " + resdata.getResCustId());
System.out.println("Phone = " + resdata.getResPhone());
System.out.println("Email = " + resdata.getResEmail());
System.out.println("Note = " + resdata.getResNote());
System.out.println("MaskedPan = " + resdata.getResMaskedPan());
System.out.println("Exp Date = " + resdata.getResExpDate());
System.out.println("Crypt Type = " + resdata.getResCryptType());
System.out.println("Avs Street Number = " + resdata.getResAvsStreetNumber());
System.out.println("Avs Street Name = " + resdata.getResAvsStreetName());
System.out.println("Avs Zipcode = " + resdata.getResAvsZipcode());
}
catch (Exception e)
{
e.printStackTrace();
}
}
```

For more details on the Vault API please download the API and Integration Guide at https://developer.moneris.com/.

3.3 Developing for Hosted Vault

- 3.3.1 Adding a new profile to Vault using the Hosted Vault
- 3.3.2 Updating a Vault Profile Using the Hosted Vault

- 3.3.3 Transaction Verification via Hosted Vault

3.3.1 Adding a new profile to Vault using the Hosted Vault

- 3.3.1.1 Required Variables - Adding New Profile to Vault
- 3.3.1.2 Optional Variables - Adding a New Profile to Vault
- 3.3.1.3 Optional 'rvar' Variables

Following are a series of tables containing all the fields that can be sent in an HVARU request while adding a new profile to the Vault. The first table contains the required variables — these must be sent to properly register a profile. Subsequent tables contain variables that can be sent optionally. The appearance and functionality of the Hosted Vault Page is controlled by the Hosted Vault Configuration Tool located in the outlined above.

3.3.1.1 Required Variables - Adding New Profile to Vault

Table 1: Required Variables - Adding New Profile to Hosted Vault

	form	https://esqa.moneris.com/HPPDP/index.php - Development https://www3.moneris.com/HPPDP/index.php - Production
res_id	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
res_key	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
cc_crypt_type	hidden	Electronic Commerce Indicator (ECI) consists of 1 digit. Possible values are: 1 - Mail Order/Telephone Order - Single 2 - Mail Order/Telephone Order - Recurring 3 - Mail Order/Telephone Order - Instalment 7 - Electronic Transaction with SSL

Below is a sample of the HVARU add account request using only the required variables.

Sample Add Profile to Vault - Required Variables

```
<FORM METHOD="POST" ACTION="https://esqa.moneris.com/HPPDP/index.php" >
<INPUT TYPE="HIDDEN" NAME="res_id" VALUE="QRZX9qa002">
<INPUT TYPE="HIDDEN" NAME="res_key" VALUE="resTTJYGTDLNB">
<INPUT TYPE="HIDDEN" NAME="cc_crypt_type" VALUE="7">
<!--MORE OPTIONAL VARIABLES CAN BE DEFINED HERE -->
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

3.3.1.2 Optional Variables - Adding a New Profile to Vault

Table 1: Optional Variables - Adding New Profile to Hosted Vault

Variable name	Type	Description

lang	hidden	This defines what language the Hosted Vault Page will be in:
		en-ca = English
		fr-ca = French
		If the tag is not included the Hosted Vault Page will default to English.

cust_id	50 alphanumeric	This is an ID field that can be used to identify the client, commonly used for student #s, policy #s, client name or invoice #s. Cannot be more than 50 characters. IT IS STRONGLY RECOMMENDED TO ALWAYS SEND A CUST_ID TO UNIQUELY IDENTIFY YOUR CLIENTS.
phone	20 alphanumeric	This is the cardholder’s phone number
email	100 alphanumeric	This is where you would include the cardholder’s email address should you wish to have an email receipt sent to them. Can not be more than 50 characters.
note	100 alphanumeric	This is any special instructions that you or the cardholder

Variable name	Type	Description
		might like to store. Can not be more than 50 characters.

NOTE: Request fields allow the following characters: a-z A-Z 0-9 _ - : . @ $ = /

The code below will set the optional fields. These are details that will identify the customer’s profile.

Sample Add Profile to Vault - Optional Variables

```
<INPUT TYPE="HIDDEN" NAME="cust_id" VALUE="invoice: 123456-12-1">
<INPUT TYPE="HIDDEN" NAME="phone" VALUE ="416 555 1212">
<INPUT TYPE="TEXT" NAME="email" VALUE="john.smith@moneris.com">
<INPUT TYPE="TEXT" NAME="note" VALUE="All deliveries go to back door">
<INPUT TYPE="HIDDEN" NAME="lang" VALUE="fr-ca">
```

3.3.1.3 Optional 'rvar' Variables

Things to Know:

- Where ***n*** is an alphanumeric value less than 10 characters long, unique to each rvar variable.
- The data sent in the rvar variables will NOT be stored in the Merchant Resource Center. These fields will be echoed backin the transaction response – in a GET or POST method.
- They may also be sent in the email receipt to the merchant if **Include 'rvar' in merchant email** is selected in the Email Receipt Configuration.

Table 1: Optional Variables - rvar

Variable name	Type	Description
rvar***n***	hidden	If these extra variables are sent in the request, they will be echoed back in the response (if GET or POST have been selected for the Response Method). Commonly used for session IDs. These variables must begin with “rvar” and then contain any alphanumeric string (i.e. rvar1, rvarname, rvarMyVariable).

The code below will send 3 rvar's in the request so that they may be returned in the response or displayed on the merchant's email receipt.

Sample Optional rvar request - Add Vault Profile

```
<INPUT TYPE="HIDDEN" NAME="rvar1" VALUE="TWO">
<INPUT TYPE="HIDDEN" NAME="rvar_monkey" VALUE="monkeys are funny">
<INPUT TYPE="HIDDEN" NAME="rvar_123" VALUE="abc">
```

3.3.2 Updating a Vault Profile Using the Hosted Vault

Below are a series of tables containing all the fields that can be sent in an HVARU request while updating an existing Vault profile. The first table contains the required variables – these must be sent to properly update the profile. Subsequent tables contain variables that can be sent optionally. The appearance and functionality of the Hosted Vault Page is controlled by the Hosted Vault Configuration Tool located in the Merchant Resource Center outlined above.

3.3.2.1 Required Variables - Updating Vault Profile

Table 1: Required Variables - Adding New Profile to Hosted Vault

Variable name	Type	Description
	form	Testing:https://esqa.moneris.com/HPPDP/index.php Production: https://www3.moneris.com/HPPDP/index.php
res_id	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
res_key	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
data_key	Hidden	Provided by Moneris Solutions in the response to a Vault Add Profile request. Identifies the unique Vault profile to be updated.
cc_crypt_type	hidden	Electronic Commerce Indicator (ECI) consists of 1 digit. Possible values are: 1 - Mail Order/Telephone Order - Single 2 - Mail Order/Telephone Order - Recurring

Variable name	Type	Description
		3 - Mail Order/Telephone Order - Instalment 7 - Electronic Transaction with SSL

Below is a sample of the HVARU update request using only the required variables.

Sample Add Profile to Vault - Required Variables

```
<FORM METHOD="POST" ACTION="https://esqa.moneris.com/HPPDP/index.php" >
<INPUT TYPE="HIDDEN" NAME="res_id" VALUE="QRZX9qa002">
<INPUT TYPE="HIDDEN" NAME="res_key" VALUE="resTTJYGTDLNB">
<INPUT TYPE="HIDDEN" NAME="data_key" VALUE="123QWERTY123qwerty">
<INPUT TYPE="HIDDEN" NAME="cc_crypt_type" VALUE="7">
<!--MORE OPTIONAL VARIABLES CAN BE DEFINED HERE -->
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

NOTE: In the update transaction only the financial details (e.g. card number, expiry date) can be altered through the Hosted Vault Registration page. All other account details must be updated through an API.

3.3.2.2 Optional 'rvar' Variables - Updating Vault Profile

Things to Know:

- Where ***n*** is an alphanumeric value less than 10 characters long, unique to each rvar variable.
- The data sent in the rvar variables will NOT be stored in the Merchant Resource Center. These fields will be echoed backin the transaction response – in a GET or POST method.
- They may also be sent in the email receipt to the merchant if **Include 'rvar' in merchant email** is selected in the Email Receipt Configuration.

Table 1: Optional Variables - rvar

Variable name	Type	Description
rvar***n***	hidden	If these extra variables are sent in the request, they will be echoed back in the response (if GET or POST have been selected for the Response Method). Commonly used for session IDs. These variables must begin with “rvar” and then contain any alphanumeric

Variable name	Type	Description
		string (i.e. rvar1, rvarname, rvarMyVariable).

Sample Update Vault Profile with rvar Variables

```
<INPUT TYPE="HIDDEN" NAME="rvar1" VALUE="TWO">
<INPUT TYPE="HIDDEN" NAME="rvar_monkey" VALUE="monkeys are funny">
<INPUT TYPE="HIDDEN" NAME="rvar_123" VALUE="abc">
```

3.3.3 Transaction Verification via Hosted Vault

- 3.3.3.1 Sending a Transaction Verification Request Via Hosted Vault below

3.3.3.1 Sending a Transaction Verification Request Via Hosted Vault

In order to perform a Transaction Verification it is essential that you configure the Hosted Vault configuration accordingly. If the Hosted Vault is properly configured you will receive a variable in a GET or POST response called "transactionKey".

It is advised that you log the initial transaction response and then compare the Transaction Verification response to ensure authenticity. The transaction verification request should be performed using server to server communication rather than sending the request through the browser.

Transaction Verification can only be performed once on a given transaction, and it can only be performed within 15 minutes of the original transaction.

The transaction verification **must** be performed using a server to server request. The verification should not be sent through the browser.

Table 1: Variables for Transaction Verification - Hosted Vault

Variable name	Type	Description
	form	https://esqa.moneris.com/HPPDP/index.php - Development https://www3.moneris.com/HPPDP/index.php- Production
res_id	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
res_key	hidden	Provided by Moneris Solutions – Hosted Vault Configuration Tool
transactionKey	Hidden	This is returned in the transaction response.

Following is a sample of the Transaction Verification Request. This is just a sample for quick testing; the verification should be performed as a server to server request.

Sample Transaction Verification Request - Hosted Vault

```
<FORM NAME=VERIFY ACTION="https://esqa.moneris.com/HPPDP/verifyTxn.php">
<INPUT TYPE="HIDDEN" NAME="ps_store_id" VALUE="qampg">
<INPUT TYPE="HIDDEN" NAME="hpp_key" VALUE="hpPu7yr4Hn5k">
<INPUT TYPE="HIDDEN" NAME="transactionKey" VALUE="SDGJSGI4ujwrlsfGSGsjhlfhs5lsffs4h">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to perform verification">
</FORM>
```

Once Moneris Gateway receives the transaction verification request we match the key, then verify and log the request. A transaction verification response is then returned with the transaction information and a status. This response is created in the format defined in the “Security Features” portion of the Hosted Payment Page configuration. Please see the following table for a list of possible Transaction Verification statuses.

Table 2: Response Fields - Transaction Verification Request - Hosted Vault

Variable name	Type	Description
data_key	50-character alphanumeric	data_key of the original transaction
response_code	3-character alphanumeric	Transaction Response Code from the original transaction < 50: Transaction approved >= 50: Transaction declined NULL: Incomplete registration - Profile registration was not attempted
transactionKey	100-character alphanumeric	The transactionKey from the request
status	alphanumeric	This is the value to check to see if the transaction has been properly validated. Below is a list of possible replies and their meaning. Valid-Registered: The account add/update was successfully validated Invalid-Reconfirmed: The

Variable name	Type	Description
		transactionKey provided has already been validated Invalid: Unable to validate request Invalid referrer URL - ??: Invalid referrer URL

Table 3: Error Codes - Transaction Verification

Code	Message/Description
991	Invalid referrer URL - <referrer url>: Referring URL does not match what is listed in the “Security Features” portion of the Hosted Payment Page configuration, validation failed. The source URL will be returned.
994	Invalid – Reconfirmed: The transaction has already been confirmed, validation failed.
995	Invalid: Not a valid confirmation request. Either the transaction doesn’t exist or the request is older than 15 minutes, validation failed.

Below are samples of valid and invalid Transaction Verification Responses displayed on our server in XML format:

Sample Transaction Verification Response for Hosted Vault - Valid	Sample Transaction Verification Response for Hosted Vault - Invalid
<?xml version="1.0" standalone="yes"?> <response> <data_key>123QWER123QWER123</data_key> <response_code>1</response_code> <status>Valid - Registered</status> <transactionKey>SADF98AF78ADSFUASDF987</transactionKey> </response>	<?xml version="1.0" standalone="yes"?> <response> <response_code>995</response_code> <status>Invalid</status > <transactionKey>SADF98AF78ADSFUASDF987</transactionKey> </response>

4 Testing a Hosted Payment Solution

4.1 How Do I Test My Solution?

A testing environment is available for you to connect to while you are integrating your site to our payment gateway. The test environment is available 24/7; however since it is a development environment we cannot guarantee 100% availability. Also, please be aware that other merchants are using the testing environment so you may see transactions, user IDs, and Hosted Payment Page configurations that you did not create.

As a courtesy to others that are testing we ask that when you are processing refunds, changing passwords and/or trying other functions that you use only the transactions/users/configurations that you created.

Using the logins in Hosted Payment Page Configuration Tool (see page 12) , you can create your own Hosted Payment Page Configuration ID and Token. You can use these to send transactions to our test environment and configure your Hosted Payment Page. Your Configuration ID and Token will be valid for 30 days. You may test as often as required.

The test environment has been designed to replicate our production environment as closely as possible. One major difference is that we are unable to send test transactions onto the production authorization network and thus issuer responses are simulated. Additionally, the requirement to emulate approval, decline and error situations dictates that we use certain transaction variables to initiate various response and error situations.

The test environment will approve and decline transactions based on the penny value of the amount field.

EXAMPLE: a transaction made for the amount of $9.00 or $1.00 will approve since the .00 penny value is set to approve in the test environment. Transactions in the test environment should not exceed $10.00. This limit does not exist in the production environment. For a list of all current test environment responses for various penny values, please see the Test Environment Penny Response table as well as the Test Environment eFraud Response table, available for download at https://developer.moneris.com

When testing you may use the following test credit card numbers with any future expiry date.

NOTE: These responses may change without notice. Moneris Solutions recommends you regularly refer to our website to check for possible changes.

Table 1: Test Card Numbers

Card Plan	Card Number
MasterCard	5454545454545454
Visa	4242424242424242
Amex	373599005095005
Diners	36462462742008

Table 2: INTERAC® Online Payment Test Card Numbers

Card Plan	Card Number
INTERAC Online Track2	3728024906540591206=01121122334455000 5268051119993326=01121122334455000000 453781122255=011211223344550000000000

NOTE:

When testing INTERAC® Online Payment you will be forwarded to the INTERAC® Online Payment merchant testing tool. A screen will appear where certain fields need to be completed.

For an approved response you will need to enter the following data in to the fields, do not alter any of the other fields:

IDEBIT_TRACK2:3728024906540591206=01121122334455000

IDEBIT_ISSNAME:RBC

IDEBIT_ISSCONF:123456

For a declined response leave the fields blank.

Click **Post to Merchant**. Do **not** click **Validate Data** — it will return validation errors.

Financial Institution	Routing Number	Account Number	Check Number
			Any number

Gift Card Test Card Numbers

For Gift Card test credentials please contact our Integration Support team at onlinepayments@moneris.com.

4.2 What Information Will I Get As a Response to My Transaction Request?

For each transaction you will receive a response message. The fields that will be included in the response are indicated in the table below.

The Receipt can be handled in two ways depending on how the “Response Method” has been configured.

1. Moneris Gateway can generate a receipt on your behalf and present it to the client. The receipt will be relatively generic in appearance and will be based on the settings from the Hosted Payment Page Configuration in the Merchant Resource Center. Please refer to What Do I Need to Include in the Receipt? (see page 102) to configure the receipt.
2. The receipt values will be sent back to the URL specified in the Hosted Payment Page Configuration settings from the Merchant Resource Center. You can then create a custom receipt or use it to initiate a secondary process. These values can be passed back appended to the URL in a query string format or as an HTTP POST.

4.2.1 Response Fields for Transaction Request

Table 1: Response Fields - Transaction Request

Variable name	Size/Type	Description
response_order_id	50-character alphanumeric	order_id specified in request or generated by Hosted Payment Page
response_code	3-character alphanumeric	Transaction Response Code < 50: Transaction approved >= 50: Transaction declined NULL: Transaction was not sent for authorization If you would like further details on the response codes that are returned please see the Response Codes document available for download at: https://developer.moneris.com

Variable name	Size/Type	Description
date_stamp	yyyy-mm-dd	Processing host date stamp
time_stamp	##:##:##	Processing host time stamp
bank_approval_code	8-character alphanumeric	Authorization code returned from the issuing institution
result	1-character numeric	1 = approved , 0 = declined, incomplete
trans_name	alphanumeric	Type of transaction that was performed purchase: cardholder was billed immediately preauth: funds were locked on the card – a capture will need to be performed to have the funds deposited into merchant's account (see Merchant Resource Centre User's Guide). A PreAuth transaction **must** be reversed if it is not to be captured. To reverse the full amount of the PreAuth, please use the Capture transaction with a dollar amount of "0.00". cavv_purchase: similar to purchase but a VbV/MCSC authentication attempt was made. cavv_preauth: similar to preauth but a VbV/MCSC authentication attempt was made. idebit_purchase: similar to purchase but the transaction was performed using INTERAC® Online Payment
cardholder	40-character alphanumeric	Cardholder's name

Variable name	Size/Type	Description
charge_total	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	Amount of the transaction

card	2-character alphanumeric	Credit Card Type
		M = Mastercard
		V = Visa
		AX = American Express
		DC = Diners Card
		NO = Novus / Discover
		C = JCB
		SE = Sears
		P = INTERAC® Online Payment

f4l4	####***####	First 4 and last 4 digits of the card #
message	100-character alphanumeric	Response description returned from issuing institution or from Moneris Gateway if there is a system error.
iso_code	2-character numeric	ISO response code
bank_transaction_id	18-character numeric	The reference number is an 18-character string that references the terminal used to process the transaction as well as the shift, batch and sequence number. This data is typically used to reference transactions on the host systems and must be displayed on any receipt presented to the customer. This information should be stored by the merchant.

Variable name	Size/Type	Description
		EXAMPLE: The following illustrates the breakdown of this field where "660123450010690030" is the reference number returned in the message, "66012345" is the terminal id, "001" is the shift number, "069" is the batch number and "003" is the transaction number within the batch.
transactionKey	100-character alphanumeric (optional)	This is an encrypted string that is returned when using the transaction verification feature. There is no need to decrypt the string. It needs to be passed back to Moneris Gateway to verify the authenticity of the transaction. **NOTE:** This variable applies only when using transaction verification functionality.
ticket	alphanumeric	The value returned from the preload data request. **NOTE:** This variable applies only when using data preload functionality.
rvarn	optional	These extra variables can be sent in the request and will be echoed back in the response. These variables must begin with "rvar" and then contain any alphanumeric string (i.e. rvar1, rvarname, rvarMyVariable). If they are not posted in the request, they will not be

<table>
<tr><th>Variable name</th><th>Size/Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>included in the response.</td></tr>
<tr><td>eci</td><td>1-character numeric</td><td>Electronic Commerce Indicator that was sent with the transaction.
Possible values are:
<table>
<tr><td>Crypt Type</td><td>TVisa/MCSC Definitions</td></tr>
<tr><td>5</td><td>- Fully authenticated
- There is a liability shift and the merchant is protected from chargebacks.</td></tr>
<tr><td>6</td><td>- VbV/MCSC has been attempted
- VbV -There is a liability shift and the merchant is protected from chargebacks
-MCSV –No liability shift and the merchant is not protected from chargebacks.</td></tr>
<tr><td>7</td><td>- Non-VbV/MCSC transaction
- Merchant is no longer protected from chargebacks</td></tr>
</table>
</td></tr>
<tr><td>txn_num</td><td>20-character alphanumeric</td><td>Gateway Transaction identifier. This value is required if merchant decides to send automated captures, voids or</td></tr>
</table>

Variable name	Size/Type	Description
		refunds through an API.
recur_result	true	Indicates the Recurring Billing result. true: The Recurring Billing transaction was successfully registered. Any response other than "true" indicates that the recurring billing transaction was not properly registered.
avs_response_code	1-character alphanumeric	Indicates the address verification result. Refer to Appendix A Transaction Request Examples for further details. To test AVS you must create a configuration in "store5" and use that configuration for testing.
cvd_response_code	1-character alphanumeric	Indicates the CVD validation result. Refer to Appendix A Transaction Request Examples for further details. To test CVD you must create a configuration in "store5" and use that configuration for testing. If you have created your own test store and it is
cavv_result_code	1-character alphanumeric	The Cardholder Authentication Verification Value (CAVV) is a value that allows VisaNet to validate the integrity of the VbV transaction data. These values are passed back from the issuer to the merchant after the VbV/SecureCode authentication has taken place. **EXAMPLE:** If the eci returned is a "6" and the result code is a "B", it becomes liable for chargeback. Please see CAVV Result Codes -

Variable name	Size/Type	Description
		Verified by Visa (see page 89) for the CAVV result codes table
is_visa_debit	boolean	A value of ‘true’ or ‘false’ is sent back which indicates if the card provided by the cardholder was a Visa Debit card.

Table 2: Response Fields - INTERAC® Online Payment

Variable name	Size/Type	Description
Trans_name	alphanumeric	Type of transaction that was performed idebit_purchase: similar to purchase but the transaction was performed using INTERAC® Online Payment
ISSNAME	1-30 characters alphanumeric	Returned for an INTERAC® Online Payment transaction. This field identifies the name of the card issuer. This data must be displayed on a receipt.
INVOICE	1-20 characters alphanumeric	Returned for an INTERAC® Online Payment transaction. This field contains the invoice number used to identify the transaction. This data must be displayed on the receipt.
ISSCONF	1-15-character alphanumeric	Returned for an INTERAC® Online Payment transaction. This field is the confirmation number returned by the issuing bank. This data must be displayed on the receipt.

Table 3: Response Fields - Gift Card Transactions

Variable name	Size/Type	Description
gift_charge_total	9-character numeric	This is the total amount of the Purchase transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
rem_balance	9-character numeric	This is the remaining balance on the card after Deactivation. The balance will be in pennies.
display_text	82-character alphanumeric	This is the remaining balance on the card after Deactivation. The balance will be in pennies.
receipt_text	122-character alphanumeric	This is a message that, if present, is to be printed on the receipt
voucher_text	255-character alphanumeric	If the VoucherType field is non-zero, the text from this field should be printed in the body of the voucher.
ref_num	10-character numeric	This is the unique number that was assigned by the Moneris system to identify the transaction. The maximum value of this parameter is 0xFFFFFFFF (4294967295). The host can not return reference numbers greater than this value. If this field is present, it is to be included on the receipt.
terminal_id	8-character numeric	Identifies the Terminal Identifier which was used to process the transaction.
txn_num	30-character alphanumeric	Gateway Transaction identifier. This value is required if merchant decides to send automated void/refund through an API.

NOTE: Multiple gift cards may be used to cover the full amount of the transaction. If two gift cards are submitted for processing, then there will be two sets of the above <gift_card> response fields within the response XML.

Table 4: Response Fields - Convenience Fee Transactions

Variable name	Size/Type	Description
convenience_fee	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	Charge the convenience fee amount. Please note the 'convenience_fee' must be less than the 'charge_total'.
cf_fee_rate	9-character decimal	The convenience fee rate that has been defined on the merchant's profile. For example: 1.00 – a fixed amount or 10.0 - a percentage amount
cf_fee_type	AMT / PCT	The type of convenience fee that has been defined on the merchant's profile. Available options are: AMT – fixed amount PCT – percentage
cf_success	true/false	Indicates whether the Convenience Fee transaction processed successfully.

Table 5: Response Fields - Transaction Risk Management Tool Transactions

Variable name	Size/Type	Description
risk_policy_score		The sum of all the risks weights from triggered rules within the selected policy in the range [-100...+100].

Variable name	Size/Type	Description
risk_request_result		success – ThreatMetrix was able to process the request successfully fail_access – ThreatMetrix was unable to process the request due to API verification failing fail_verification – API query limit reached fail_incomplete – ThreatMetrix was unable to process the request due to incomplete or incorrect input data fail_internal_error – ThreatMetrix encountered an error while processing the request fail_temporarily_unavailable – the request fail because the service is temporarily unavailable fail_invalid_email_address – the format of the supplied email address was invalid fail_invalid_telephone_number – the format of the supplied telephone number was invalid fail_invalid_device_id – the format of the supplied device_id was invalid fail_invalid_ip_address_parameter – the format of a supplied ip_address parameter was invalid
risk_reason_code		The codes of the rules verified from the selected policy that have triggered. Each rule code is returned as a separate name/value pair.

Variable name	Size/Type	Description
risk_reason_name		The names of rules verified from the selected policy that have triggered. Each rule name is returned as a separate name/value pair.
risk_reason_message_en		An English message description of the rule returned.
risk_reason_message_fr		A French message description of the rule returned.

Table 6: Response Fields - Loyalty Card Transactions

Variable name	Type	Description
request_amount	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	Identifies the amount for which loyalty points are to be awarded. This amount may be equal to or less than the total amount of the transaction.
transaction_points	9-character numeric	Amount processed on this loyalty card transaction. This value will be displayed in the number of points.
transaction_amount	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	Amount processed on this loyalty card transaction. This value will be displayed in the number of points.
current_balance	9-character numeric	This is the current balance for the card in points. If this field is present, it is to be printed on the receipt. If this field is not present, no balance information is to be printed on the receipt
lifetime_balance	9-character numeric	This is the lifetime balance for the card in points. If this field is

Variable name	Type	Description
		present, it is to be printed on the receipt. If this field is not present, no balance information is to be printed on the receipt.

Table 7: Response Fields - Hosted Vault Transactions

Variable name	Type	Description
data_key	max 50-character alphanumeric	The unique key to identify the client. This is the ID that will be used for subsequent transactions for the account, such as an update.
payment_type	alphanumeric	This identifies what type of payment was registered. Possible values are:
res_success	4-character alphanumeric	True: Card registered False: Card failed registration Null: Incomplete registration

NOTE: To determine if a transaction is approved, the response_code will have a value of less than 50. If it is declined the response_code will be 50 or greater. A value of NULL means the transaction was incomplete.

4.2.2 Special Error Codes

The Hosted Payment Page is designed to generate special error codes when certain data is incorrect and/or the transaction couldn’t be processed. The table below contains the information regarding the error codes. Each error will be accompanied by a message describing the problem.

Table 1: Special Error Codes

Code	Message/Description
914	Transaction cancelled by cardholder – The response code indicates that the cardholder pressed the <cancel> transaction button – This response is only returned if the enhanced cancel button functionality is enabled within the Hosted Paypage configuration.
991	Invalid referrer URL - <referrer url> – If the Hosted Payment Page solution is configured to check the referring URL and it is incorrect this error will occur. The source URL will be included in the error. Please refer to the "Security Features" portion of the Hosted Payment Page configuration for a list of all Allowed Referring URLs.
992	VbV / Secure Code authentication failed – This error will occur if your merchant account is configured for VbV/MCSC and the cardholder failed to enter the proper PIN during the authentication process.
993	Data error - unable to store data – This error will occur if too much request data was passed in the transaction request or if the database failed to store the request. This may occur if unsupported characters were included in one of the posted fields.
N/A	Invalid store credentials – There is no code generated and a blank page is loaded with the above information. The ps_store_id and/or hpp_key did not match an existing store.
N/A	Card Issuer returned corrupt data. Unable to proceed with the transaction. Please return to the site where you initiated the transaction and try again. Your card has not been charged. – There is no code generated and a blank page is loaded with the above information. This error will occur if the cardholder's issuing bank did not return the correct data in the VbV/MCSC authentication process.

Table 2: Error Codes - Convenience Fee Responses

Code	Message/Description
973	Unable to locate merchant CF details

Code	Message/Description
977	Invalid amount
978	Failed CF transaction
984	Data error: (optional: field name)
987	Invalid transaction
Null	Error: Malformed XML

4.3 Understanding the Fraud Prevention Tools

4.3.1 Address Verification Service (AVS)

The Address Verification Service (AVS) value refers to the cardholder’s street number, street name and zip/postal code as it would appear on their statement. When participating in this security feature the Hosted Payment Page will prompt the cardholder for the AVS information.

4.3.2 Card Validation Digit (CVD)

The Card Validation Digit (CVD) value refers to the numbers appearing on the back of the credit card which are not imprinted on the front. The exception to this is with American Express card where this value is indeed printed on the front. When participating in this security feature the Hosted Payment Page will prompt the cardholder to enter CVD value.

4.3.3 Verified by Visa (VbV)

Verified by Visa (VbV) is a program offered by Visa. Before approving a transaction Moneris Gateway and the Bank that issues the credit cards will attempt to authenticate the cardholder through the use of a password, similar to a debit PIN. Merchants who have enrolled in these programs with Moneris will be able to offer their customers added protection against unauthorized credit card use, as well as protect themselves from fraud-related chargebacks.

If you have enrolled in Verified by Visa (VbV) with Moneris, the Hosted Payment Page will automatically attempt to perform the VbV authentications.

4.3.4 MasterCard SecureCode (MCSC)

MasterCard SecureCode (MCSC) is a new feature offered by MasterCard. Merchants who have enrolled in this program with Moneris and Moneris Gateway will be able to offer their customers added protection against unauthorized credit card use, as well as protect themselves from fraud-related chargebacks. Cardholders that have applied for SecureCode with their issuing bank will be able to use this password similar to a debit PIN number for online transactions with participating online merchants.

Before approving a transaction, Moneris Gateway and the Bank that issued the MasterCard will authenticate the cardholder through the use of this password. For merchants who have enrolled in SecureCode, the Hosted Payment Pagesolution will automatically attempt to perform SecureCode verification on every MasterCard transaction.

4.3.5 Transaction Risk Management Tool

The Transaction Risk Management Tool provides additional information to assist in identifying fraudulent transactions. For merchants who have enrolled in the Transaction Risk Management Tool, the Hosted Paypage can be configured to obtain a risk assessment for every transaction and return the results along with the transaction response information.

4.3.6 How Do I Handle the eFraud Response Information?

When reviewing the response information and determining how to handle the transaction, it is recommended that you (either manually or through automated logic on your site) use the following pieces of information:

1. The risk score
2. The rules triggered (e.g. Rule Codes, Rule Names, Rule Messages) Results obtained from Verified by Visa, MasterCard Secure Code, AVS, CVD and the financial transaction authorization
3. Automated processes will also need to include the response codes for the Transaction Risk Management Transaction

4.3.6.1 Card Validation Digits (CVD) and Address Verification Service (AVS)

Card Validation Digits (CVD)

The Card Validation Digits (CVD) value refers to the numbers appearing on the **back** of the credit card. The exception to this is with American Express cards where this value is printed on the front

Address Verification Service (AVS)

The Address Verification Service (AVS) value refers to the cardholder's street number, street name and zip/postal code as it would appear on their statement.

Additional Information for CVD and AVS

The responses that are received from CVD and AVS verifications are intended to provide added security and fraud prevention, but the response itself will not affect the issuer's approval of a transaction. Upon receiving a response, the choice to proceed with a transaction is left entirely to the merchant.

Please note that all responses coming back from these verification methods are not direct indicators of whether a merchant should complete any particular transaction. The responses should not be used as a strict guideline of which transaction will approve or decline.

NOTE:

CVD verification is only applicable towards Visa, MasterCard and American Express transactions.

Also, please note that AVS verification is only applicable towards Visa, MasterCard, Discover and American Express transactions. This verification method is not applicable towards any other card type.

*For additional information on how to handle these responses, please refer to the eFraud (CVD & AVS) Result Codes document which is available at https://developer.moneris.com

Below is a sample of the AVS and CVD response displayed on our server in XML format.

Sample AVS and CVD Response - Valid

```
<?xml version="1.0" standalone="yes"?>
<response>
      <order_id>mhp2250927670</order_id>
<response_code>27</response_code>
<amount>10.10</amount>
<txn_num>135172-0_77</txn_num>
<avs_response_code>A</avs_response_code>
<cvd_response_code>M</avs_response_code>
</response>
```

4.3.6.2 CAVV and Crypt Types

The Cardholder Authentication Verification Value (CAVV) is a value that allows validation of the integrity of Verified by Visa (VbV), MasterCard SecureCode (MCSC) or American Express SafeKey authentication data. This value is passed from the Issuer to the merchant after the authentication has taken place. The Hosted Payment Page then integrates the CAVV value into the financial authorization request to the Issuer.

The crypt type is derived by Moneris Gateway using the CAVV returned during authentication using VbV, MCSC or SafeKey. Below are the possible values returned.

Crypt Type	Visa, MasterCard and AmericanExpress Definition
5	- Fully authenticated - There is a liability shift and the merchant is protected from chargebacks
6	- VbV/MCSC/Safekey has been attempted - There is a liability shift and the Merchant is protected from chargebacks of certain types

Crypt Type	Visa, MasterCard and AmericanExpress Definition
7	- Non-VbV/MCSC/Safekey transaction - No liability shift - Merchant is no longer protected from chargebacks

CAVV Result Codes - Verified by Visa

NOTE: This information applies to Verified by Visa transactions only. It does not apply to MCSC or SafeKey transactions.

The Cardholder Authentication Verification Value (CAVV) is a value that allows VisaNet to validate the integrity of the VbV authentication data. This value is passed from the Issuer to the merchant after the VbV authentication has taken place. The Hosted Payment Page then integrates the CAVV value into the financial authorization request to the Issuer.

Once the VbV authentication is completed and the financial authorization transaction (Purchase or Auth) has been authorized, the CAVV Result Code value may be returned in the financial transaction response to provide the merchant with additional details pertaining to the integrity of the VbV validation.

The following table describes the content of the CAVV Result Code response data and what it means to the merchant.

Table 1: CAVV Result Codes

Result Code	Message	What this means to you as a merchant...
0	CAVV authentication results invalid.	For this transaction you may not receive protection from chargebacks as a result of using VbV as the CAVV was considered invalid at the time the financial transaction was processed. Please check that you are following the VbV process correctly and passing the correct data in our transactions.

Result Code	Message	What this means to you as a merchant…
1	CAVV failed validation; authentication	Provided that you have implemented the VbV process correctly the liability for this transaction should remain with the Issuer for chargeback reason codes covered by Verified by Visa.
2	CAVV passed validation; authentication	The CAVV was confirmed as part of the financial transaction. This transaction is a fully authenticated VbV transaction (ECI 5)
3	CAVV passed validation; attempt	The CAVV was confirmed as part of the financial transaction. This transaction is an attempted VbV transaction (ECI 6)
4	CAVV failed validation; attempt	Provided that you have implemented the VbV process correctly the liability for this transaction should remain with the Issuer for chargeback reason codes covered by Verified by Visa.
7	CAVV failed validation; attempt (US issued cards only)	Please check that you are following the VbV process correctly and passing the correct data in our transactions. Provided that you have implemented the VbV process correctly the liability for this transaction should be the same as an attempted transaction (ECI 6)
8	CAVV passed validation; attempt (US issued cards only	The CAVV was confirmed as part of the financial transaction. This transaction is an attempted VbV transaction (ECI 6)

Result Code	Message	What this means to you as a merchant…
9	CAVV failed validation; attempt (US issued cards only)	Please check that you are following the VbV process correctly and passing the correct data in our transactions. Provided that you have implemented the VbV process correctly the liability for this transaction should be the same as an attempted transaction (ECI 6)
A	CAVV passed validation; attempt (US issued cards only)	The CAVV was confirmed as part of the financial transaction. This transaction is an attempted VbV transaction (ECI 6)
B	CAVV passed validation	The CAVV was confirmed as part of the financial transaction. However, this transaction doesn't qualify for the liability shift. Treat this transaction the same as an ECI 7.

4.3.6.3 Transaction Risk Management Tool Responses

The responses that are received from CVD and AVS verifications are intended to provide added security and fraud prevention, but the response itself will not affect the completion of a transaction. Upon receiving a response, the choice to proceed with a transaction is left entirely to the merchant.

The responses that are received from the Transaction Risk Management Tool are intended to provide added security and fraud prevention, but the response itself will not affect the completion of a transaction. Upon receiving a response, the choice to proceed with a transaction is left entirely to the merchant.

Below is a sample of the Transaction Risk Management Tool response displayed on our server in XML format.

Sample Transaction Risk Management Tool Response - XML

```
<?xml version="1.0" standalone="yes"?>
<response>
      <order_id>mhp2250927670</order_id>
<response_code>27</response_code>
<amount>5.00</amount>
<txn_num>135172-0_77</txn_num>
```

Sample Transaction Risk Management Tool Response - XML

```
      <risk_policy_score>-15</risk_policy_score>
<risk_request_result>success</risk_request_result>
    <risk_reason_code>UN001,UN001</risk_reason_code>
    <risk_reason_name> 3DevciePerAccountNumberWeek,3DevciePerCreditCardWeek</risk_reason_name>
    <risk_reason_message_en> Unknown Rule,Unknown Rule</risk_reason_message_en>
    <risk_reason_message_fr> Regle Inconnus,Regle Inconnus<risk_reason_message_fr>
</response>
```

Understanding the Risk Score

For each transaction through the Hosted Payment Page with the Transaction Risk Management Tool configured, a score with a value between -100 and +100 will be returned based on the rules that were triggered for the transaction. Below is a table defining the different possible risk scores ranges.

Table 1: Risk Score Definitions

Risk Score	Visa Definition
[-100 … -1]	The lowest score that can be reached is -100. The more negative the number (ie closer to -100) the more likely the transaction is fraudulent.
0	A risk score of 0 indicates a neutral transaction
[1 … +100]	The highest score that can be reached is +100. The more positive the number (ie closer to +100) the lower the risk that the transaction is fraudulent. **NOTE:** All e-commerce transactions have some level of risk associated with them and as a result it is rare to see transactions with a risk score in the high positive values.

When evaluating the risk of a transaction, the risk score will give you an initial indicator of the potential risk level that the transaction is or isn't fraudulent. The more negative the score, the higher the probability is that the transaction is fraudulent. Since some of the rules that are evaluated on each transaction may/may not be as relevant in your business scenario, you should also review the rules that were triggered for the transaction before determining how to handle the transaction.

Understanding the Rule Codes, Rule Names and Rule Messages

The rule codes, rule names and rule messages provide details on what rules were trigged during the assessment of the information provided in Transaction Risk Management Tool. Each Rule Code has a Rule Name and Message. The Rule Name and Rule Message will typically be very similar to the table inTransaction Risk Management Tool Rules & Codes (see page 93)

When evaluating the risk of a transaction, it is recommended that you review the rules that were triggered for the transaction and assess the relevancy of it to your business (e.g. how it relates to the typical buying habits of your customer base).

If you are automating some or all of the decision making process related to handling the responses, you may want to use the Rule Codes. If you are documenting manual processes you may want to refer to the more user friendly Rule Name and/or Rule Message.

Transaction Risk Management Tool Rules & Codes

The following is a list of all possible responses of Rule Names once a Query has been performed.

Table 1: Rule Number and Rule Description

Rule Name	Rule Number	Message/Description	Rule Explanation
White lists			
DeviceWhitelisted	WL001	Device White Listed	Device is on the white list. This indicates that the device has been flagged as always "ok". NOTE: This rule is currently not in use.
IPWhitelisted	WL002	IP White Listed	IP Address is on the white list. This indicates the device has been flagged as always "ok". NOTE: This rule is currently not in use.

Rule Name	Rule Number	Message/Description	Rule Explanation
EmailWhitelisted	WL003	Email White Listed	Email address is on the white list. This indicates that the device has been flagged as always "ok". NOTE: This rule is currently not in use.
Event Velocity			
2DevicePayment	EV003	2 Device Payment Velocity	Multiple payments were detected from this device in the past 24 hours.
2IPPaymentVelocity	EV006	2 IP Payment Velocity	Multiple payments were detected from this IP within the past 24 hours.
2ProxyPaymentVelocity	EV008	2 Proxy Payment Velocity	The device has used 3 or more different proxies during a 24 hour period. This could be a risk or it could be someone using a legitimate corporate proxy.
Email			
3EmailPerDeviceDay	EM001	3 Emails for the Device ID in 1 Day	This device has presented 3 different email ids within the past 24 hours.
3EmailPerDeviceWeek	EM002	3 emails for the Device ID in 1 week	This device has presented 3 different email ids within the past week.
3DevciePerEmailDay	EM003	3 Device Ids for email address in 1 day	This email has been presented from three different devices in the past 24 hours.

Rule Name	Rule Number	Message/Description	Rule Explanation
3DevciePerEmailWeek	EM004	3 Device Ids for email address in 1 week	This email has been presented from three different devices in the past week.
EmailDistanceTravelled	EM005	Email Distance Travelled	This email address has been associated with different physical locations in a short period of time.
3EmailPerSmartIDHour	EM006	3 Emails for SmartID in 1 Hour	The SmartID for this device has been associated with 3 different email addresses in 1 hour.
GlobalEMailOverOneMonth	EM007	Global Email over 1 month	The e-mail address involved in the transaction over 30 days ago. This generally indicates that the transaction is less risky. Note: This rule is currently set currently set so it does not impact the policy score or risk rating.
ComputerGeneratedEmailAddress	EM008	Computer Generated Email Address	This transaction used a computer generated email address.
Account Number			
3AccountNumberPerDeviceDay	AN001	3 Account Numbers for device in 1 day	This device has presented 3 different user accounts within the past 24 hours.
3AccountNumberPerDeviceWeek	AN002	3 Account Numbers for device in 1 week	This device has presented 3 different user accounts within the past week.

Rule Name	Rule Number	Message/Description	Rule Explanation
3DevciePerAccountNumberDay	AN003	3 Device IDs for account number in 1 day	This user account been used from three different devices in the past 24 hours.
3DevciePerAccountNumberWeek	AN004	3 Device IDs for account number in 1 week	This card number has been used from three different devices in the past week.
AccountNumberDistanceTravelled	AN005	Account Number distance travelled	This card number has been used from a number of physically different locations in a short period of time.
Credit Card / Payments			
3CreditCardPerDeviceDay	CP001	3 credit cards for device in 1 day	This device has used three credit cards within 24 hours.
3CreditCardPerDeviceWeek	CP002	3 credit cards for device in 1 week	This device has used three credit cards within 1 week.
3DevicePerCreditCardDay	CP003	3 device ids for credit card in 1 day	This credit card has been used on three different devices in 24 hours.
3DevciePerCreditCardWeek	CP004	3 device ids for credit card in 1 week	This credit card has been used on three different devices in 1 week.
CredtCardDistanceTravelled	CP005	Credit Card has travelled	The credit card has been used at a number of physically different locations in a short period of time.
CreditCardShipAddressGeoMismatch	CP006	Credit Card and Ship Address do not match	The credit card was issued in a region different from the Ship To Address information provided.

Rule Name	Rule Number	Message/Description	Rule Explanation
CreditCardBillAddressGeoMismatch	CP007	Credit Card and Billing Address do not match	The credit card was issued in a region different from the Billing Address information provided.
CreditCardDeviceGeoMismatch	CP008	Credit Card and device location do not match	The device is located in a region different from where the card was issued.
CreditCardBINShipAddressGeoMismatch	CP009	Credit Card issuing location and Shipping address do not match	The credit card was issued in a region different from the Ship To Address information provided.
CreditCardBINBillAddressGeoMismatch	CP010	Credit Card issuing location and Billing address do not match	The credit card was issued in a region different from the Billing Address information provided.
CreditCardBINDeviceGeoMismatch	CP011	Credit Card issuing location and location of the device do not match	The device is located in a region different from where the card was issued.
TransactionValueDay	CP012	Daily Transaction Value Threshold	The transaction value exceeds the daily threshold.
TransactionValueWeek	CP013	Weekly Transaction Value Threshold	The transaction value exceeds the weekly threshold.
Proxy Rules			
3ProxyPerDeviceDay	PX001	3 Proxy Ips in 1 day	This device has used three different proxy servers in the past 24 hours.
AnonymousProxy	PX002	Anonymous Proxy IP	This device is using an anonymous proxy

Rule Name	Rule Number	Message/Description	Rule Explanation
UnusualProxyAttributes	PX003	Unusual Proxy Attributes	This transaction is coming from a source with unusual proxy attributes.
AnonymousProxy	PX004	Anonymous Proxy	This device is connecting through an anonymous proxy connection.
HiddenProxy	PX005	Hidden Proxy	This device is connecting via a hidden proxy server.
OpenProxy	PX006	Open Proxy	This transaction is coming from a source that is using an open proxy.
TransparentProxy	PX007	Transparent Proxy	This transaction is coming from a source that is using a transparent proxy.
DeviceProxyGeoMismatch	PX008	Proxy and True GEO Match	This device is connecting through a proxy server that didn't match the devices geolocation.
ProxyTrueISPMismatch	PX009	Proxy and True ISP Match	This device is connecting through a proxy server that doesn't match the true IP address of the device.
ProxyTrueOrganizationMismatch	PX010	Proxy and True Org Match	The Proxy information and True ISP information for this source do not match.
DeviceProxyRegionMismatch	PX011	Proxy and True Region Match	The proxy and device region location information do not match.

Rule Name	Rule Number	Message/Description	Rule Explanation
ProxyNegativeReputation	PX012	Proxy IP Flagged Risky in Reputation Network	This device is connecting from a proxy server with a known negative reputation.
SatelliteProxyISP	PX013	Satellite Proxy	This transaction is coming from a source that is using a satellite proxy.
GEO			
DeviceCountriesNotAllowed	GE001	True GEO in Countries Not Allowed blacklist	This device is connecting from a high-risk geographic location.
DeviceCountriesNotAllowed	GE002	True GEO in Countries Not Allowed (negative whitelist)	The device is from a region that is not on the whitelist of regions that are accepted.
DeviceProxyGeoMismatch	GE003	True GEO different from Proxy GEO	The true geographical location of this device is different from the proxy geographical location.
DeviceAccountGeoMismatch	GE004	Account Address different from True GEO	This device has presented an account billing address that doesn't match the devices geolocation.
DeviceShipGeoMismatch	GE005	Device and Ship Geo mismatch	The location of the device and the shipping address do not match.
DeviceShipGeoMismatch	GE006	Device and Ship Geo mismatch	The location of the device and the shipping address do not match.
Device			
SatelliteISP	DV001	Satellite ISP	This transaction is from a source that is using a satellite ISP.

Rule Name	Rule Number	Message/Description	Rule Explanation
MidsessionChange	DV002	Session Changed Mid-session	This device changed session details and identifiers in the middle of a session.
LanguageMismatch	DV003	Language Mismatch	The language of the user does not match the primary language spoken in the location where the True IP is registered.
NoDeviceID	DV004	No Device ID	No device ID was available for this transaction.
Dial-upConnection	DV005	Dial-up connection	This device uses a less identifiable dial-up connection.
DeviceNegativeReputation	DV006	Device Blacklisted in Reputational Network	This device has a known negative reputation as reported to the fraud network.
DeviceGlobalBlacklist	DV007	Device on the Global Black List	This device has been flagged on the global blacklist of known problem devices.
DeviceCompromisedDay	DV008	Device compromised in last day	This device has been reported as compromised in the last 24 hours.
DeviceCompromisedHour	DV009	Device compromised in last hour	This device has been reported as compromised in the last hour.
FlashImagesCookiesDisabled	DV010	Flash Images Cookies Disabled	Key browser functions/identifiers have been disabled on this device.
FlashCookiesDisabled	DV011	Flash Cookies Disabled	Key browser functions/identifiers have been disabled on this device.

Rule Name	Rule Number	Message/Description	Rule Explanation
FlashDisabled	DV012	Flash Disabled	Key browser functions/identifiers have been disabled on this device.
ImagesDisabled	DV013	Images Disabled	Key browser functions/identifiers have been disabled on this device.
CookiesDisabled	DV014	Cookies Disabled	Key browser functions/identifiers have been disabled on this device.
DeviceDistanceTravelled	DV015	Device Distance Travelled	The device has been used from multiple physical locations in a short period of time.
PossibleCookieWiping	DV016	Cookie Wiping	This device appears to be deleting cookies after each session.
PossibleCookieCopying	DV017	Possible Cookie Copying	This device appears to be copying cookies.
PossibleVPNConnection	DV018	Possibly using a VPN Connection	This device may be using a VPN connection

Pulling All the Information Together to Make a Decision

Depending on your business policies and processes, you will use the information obtained from the Fraud Tools (e.g. AVS, CVD, VbV/SecureCode and Transaction Risk Management) to make an informed decision on whether you want to accept the transaction or consider it to be a potential fraudulent transaction that you do not want to continue to process.

If you do not want to continue with a transaction because it appears too risky and is likely fraudulent, you will need to:

- Let the customer know that you will not be proceeding with their order.
- Cancel the financial transaction if you have received an approved authorization. To do this you will need to send a Void/Refund for a purchase transaction or a $0.00 Capture transaction if the original transaction was a pre-authorization.

4.4 What Do I Need to Include in the Receipt?

Visa and MasterCard expect certain variables be returned to the cardholder and presented as a receipt when a transaction is approved. If the Hosted Payment Page is configured to return the response to your webserver it is imperative that you display the information listed below. These required fields are listed below and includes the corresponding variable name as returned by the Hosted Payment Page or a proper description in brackets.

1. Amount - (charge_total)
2. Transaction Type - (trans_name)
3. Convenience Fee Amount – (convenience_fee – required only for Convenience Fee)
4. Date and Time - (date_stamp & time_stamp)
5. Authorisation Code - (bank_approval_code)
6. ResponseCode - (response_code)
7. ISO Code - (iso_code)
8. Response Message - (message)
9. Reference Number - (bank_transaction_id)
10. Goods and Services Order - (description of the products / services ordered)
11. Merchant Name - (Your Business Name – should be same as what you registered with Moneris Solutions)
12. Merchant URL - (Your business website)
13. Cardholder Name - (cardholder)
14. Return Policy (only a requirement for e-commerce transactions)

The following are required for INTERAC® Online Payment **only**:

1. Issuer Name (ISSNAME)
2. Issuer Confirmation (ISSCONF)
3. Invoice Number (INVOICE)

5 Moving to Production

- 5.1 How Do I Activate My Store?
- 5.2 How Do I Configure My Store for Production?

Once you have completed the necessary steps of creating a profile for your solution, configuring the solution profile, developing and testing the solution, you are ready to move your solution into production.

5.1 How Do I Activate My Store?

Once you have received your activation letter/fax go to https://www3.moneris.com/connect/en/activate/index.php as instructed in the letter/fax. You will need to input your store ID and merchant ID then click on **Activate**. Once this is confirmed you will need to create an administrator account that you will use to log into the Merchant Resource Center to access and administer your Moneris Gateway store.

NOTE: The API TOKEN that you receive during Activation is NOT the token that you require for the Hosted Payment Page request.

5.2 How Do I Configure My Store for Production?

Once you have activated your store, the next step is to point your store to the production host.

To point your store to the production host:

1. In your HTML FORM POST, change the <FORM METHOD=”POST” ACTION=https://esqa.moneris.com/HPPDP/index.php> to contain the production URL: <FORM METHOD=”POST” ACTION=https://www3.moneris.com/HPPDP/index.php>.
2. Change the ps_store_id and hpp_key to reflect your production store configuration.

Once you are in production you will access the Merchant Resource Center at https://www3.moneris.com/mpg. You can use the store administrator ID you created during the activation process and then create additional users as needed.

For INTERAC® Online Payment Solution:

Third-party Service/Shopping-cart Provider

In your product documentation, please ensure that the clients are properly instructed to create the Production HPP configuration as outlined above. They should also be instructed to provide Moneris Solutions with screen-shots of their check-out process showing examples of approved and declined transactions using the INTERAC® Online Payment service and provide the completed Merchant Checklist of Appendix B. Detailed descriptions of the requirements for the checklist can be found in the INTERAC Online Merchant Guidelines document. Once completed, they can be faxed or emailed to the Moneris Gateway Integration Support group for review.

When you are ready to move into production please contact the Integration Support Team at eproducts@moneris.com.

You will be required to email or fax a completed copy of the Merchant Checklist found in Appendix B of this document, as well as a screenshots of your check-out process, to 1-416-734-1059 (ATTN: Moneris Gateway Integration).

Appendix A Transaction Request Examples

Transaction Requests

The example below will send both shipping and billing address information as well as item information, and initiate a recurring charge.

```
<FORM ACTION="https://esqa.moneris.com/HPPDP/index.php" method=post>
<!-- Store Settings-->
<INPUT TYPE="HIDDEN" NAME="ps_store_id" VALUE="qampg">
<INPUT TYPE="HIDDEN" NAME="hpp_key" VALUE="hpPu7yr4Hn5k">
<!------- DEFINE CHARGE TOTAL HERE --->
Amount:<INPUT TYPE="text" NAME="charge_total" VALUE="300.00"><br><br>
<!-- Unique Order ID -->
<INPUT TYPE="hidden" NAME="order_id" VALUE="hpp_mr_test_1">

<!-- Additional Optional Details -->
<input type="hidden" name="cust_id" value="customer num">
<input type="hidden" name="email" value="">
<input type="hidden" name="note" value="these are special instructions">
<!-- Item Information -->
<input type="hidden" name="quantity1" value="3">
<input type="hidden" name="description1" value="qunat 3">
<input type="hidden" name="id1" value="sku123">
<input type="hidden" name="price1" value="4.00">
<input type="hidden" name="subtotal1" value="12.00">
<input type="hidden" name="quantity2" value="2">
<input type="hidden" name="description2" value="qunat 2">
<input type="hidden" name="id2" value="2sku123">
<input type="hidden" name="price2" value="24.00">
<input type="hidden" name="subtotal2" value="212.00">
<input type="hidden" name="gst" value="3.03">
<input type="hidden" name="shipping_cost" value="4.03">
<!-- rvar Information -->
<input type="hidden" name="rvar1" value="1_rvar">
<input type="hidden" name="rvar2" value="2_rvar">
<input type="hidden" name="rvar3" value="3_rvar">
<input type="hidden" name="rvar4" value="4_rvar">
<!-- Shipping information -->
<input type="hidden" name="ship_first_name" value="sfn">
<input type="hidden" name="ship_last_name" value="sln">
<input type="hidden" name="ship_company_name" value="scn">
<input type="hidden" name="ship_address_one" value="sao">
<input type="hidden" name="ship_city" value="sc">
<input type="hidden" name="ship_state_or_province" value="ssop">
<input type="hidden" name="ship_postal_code" value="spc">
<input type="hidden" name="ship_country" value="scount">
<input type="hidden" name="ship_phone" value="sp">
<input type="hidden" name="ship_fax" value="sf">
<!-- Billing Information -->
<input type="hidden" name="bill_first_name" value="bfn">
<input type="hidden" name="bill_last_name" value="bln">
<input type="hidden" name="bill_company_name" value="bcn">
<input type="hidden" name="bill_address_one" value="bao">
<input type="hidden" name="bill_city" value="bc">
<input type="hidden" name="bill_state_or_province" value="bsop">
<input type="hidden" name="bill_postal_code" value="bpc">
<input type="hidden" name="bill_country" value="bcount">
<input type="hidden" name="bill_phone" value="bp">
<input type="hidden" name="bill_fax" value="bf">
```

```
<!-- Recurring Information -->
<input type="hidden" name="doRecur" value ="1">
<input type="hidden" name="recurUnit" value ="day">
<input type="hidden" name="recurStartDate" value="2006/06/01">
<input type="hidden" name="recurNum" value="99">
<input type="hidden" name="recurStartNow" value="true">
<input type="hidden" name="recurPeriod" value="4">
<input type="hidden" name="recurAmount" value="4.00">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

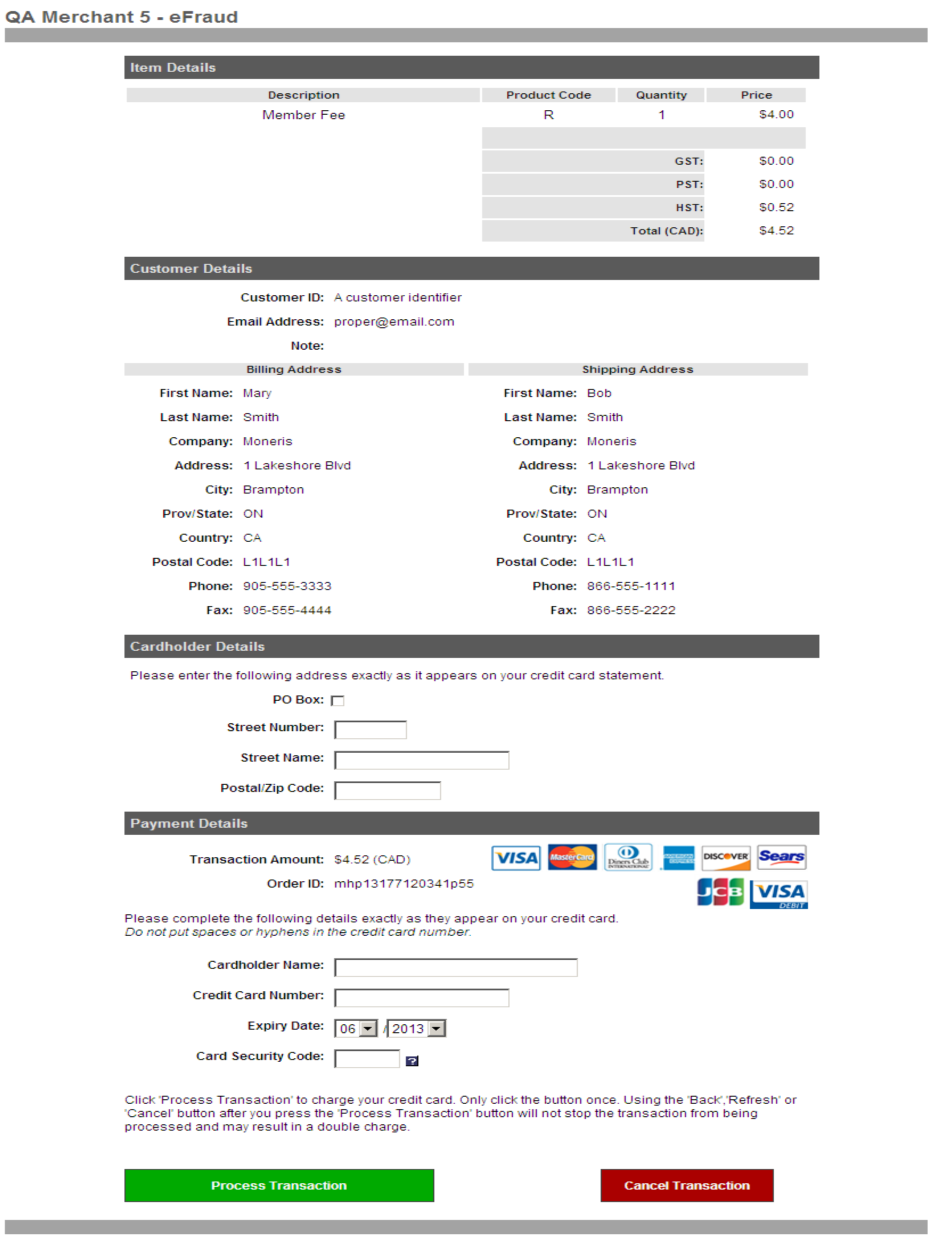
Hosted Vault Transaction Request

```
<FORM ACTION="https://esqa.moneris.com/HPPDP/index.php" METHOD="POST">
<!-- Mandatory Fields -->
<INPUT TYPE="hidden" NAME="res_id" VALUE="QRZX9qa002">
<INPUT TYPE="hidden" NAME="res_key" VALUE="reTTJYGTDLNB">
<!-- Mandatory for update ONLY, not used in this example -->
<!-- <INPUT TYPE="hidden" NAME="data_key" VALUE="58t203kF71u9dm6g75P2hA1">-->
<!-- Additional Optional Details -->
<INPUT TYPE="hidden" NAME="email" VALUE="john.smith@example.com">
<INPUT TYPE="hidden" NAME="note" VALUE="This is a note.">
<INPUT TYPE="hidden" NAME="cust_id" VALUE="Some Customer Number">
<INPUT TYPE="hidden" NAME="phone" VALUE="416 555 1212">
<INPUT TYPE="HIDDEN" NAME="lang" VALUE="fr-ca">
<!—- Payment Type information -->
<INPUT TYPE="hidden" NAME="cc_crypt_type" VALUE="7">

<!-- rvar Information -->
<INPUT TYPE="hidden" NAME="rvar_1" VALUE="1">
<INPUT TYPE="hidden" NAME="rvar_monkey" VALUE="monkeys are funny">
<INPUT TYPE="hidden" NAME="rvar_123abc" VALUE="123abc">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

Appendix B Sample Hosted Payment Page Layout

The sample below displays the layout of the Hosted Payment Page. The sections and sub-sections displayed, as well as the colour and style of the page, are determined by the settings chosen in the Hosted Payment Page Configuration. Please refer to Configuring the Hosted Payment Page (see page 15) for further information.



QA Merchant 5 - eFraud

Item Details

Description	Product Code	Quantity	Price
Member Fee	R	1	$4.00
		GST:	$0.00
		PST:	$0.00
		HST:	$0.52
		Total (CAD):	$4.52

Customer Details

Customer ID: A customer identifier
Email Address: proper@email.com
Note:

Billing Address		Shipping Address	
First Name:	Mary	First Name:	Bob
Last Name:	Smith	Last Name:	Smith
Company:	Moneris	Company:	Moneris
Address:	1 Lakeshore Blvd	Address:	1 Lakeshore Blvd
City:	Brampton	City:	Brampton
Prov/State:	ON	Prov/State:	ON
Country:	CA	Country:	CA
Postal Code:	L1L1L1	Postal Code:	L1L1L1
Phone:	905-555-3333	Phone:	866-555-1111
Fax:	905-555-4444	Fax:	866-555-2222

Cardholder Details

Please enter the following address exactly as it appears on your credit card statement.

PO Box: □
Street Number:
Street Name:
Postal/Zip Code:

Payment Details

Transaction Amount: $4.52 (CAD)
Order ID: mhp13177120341p55

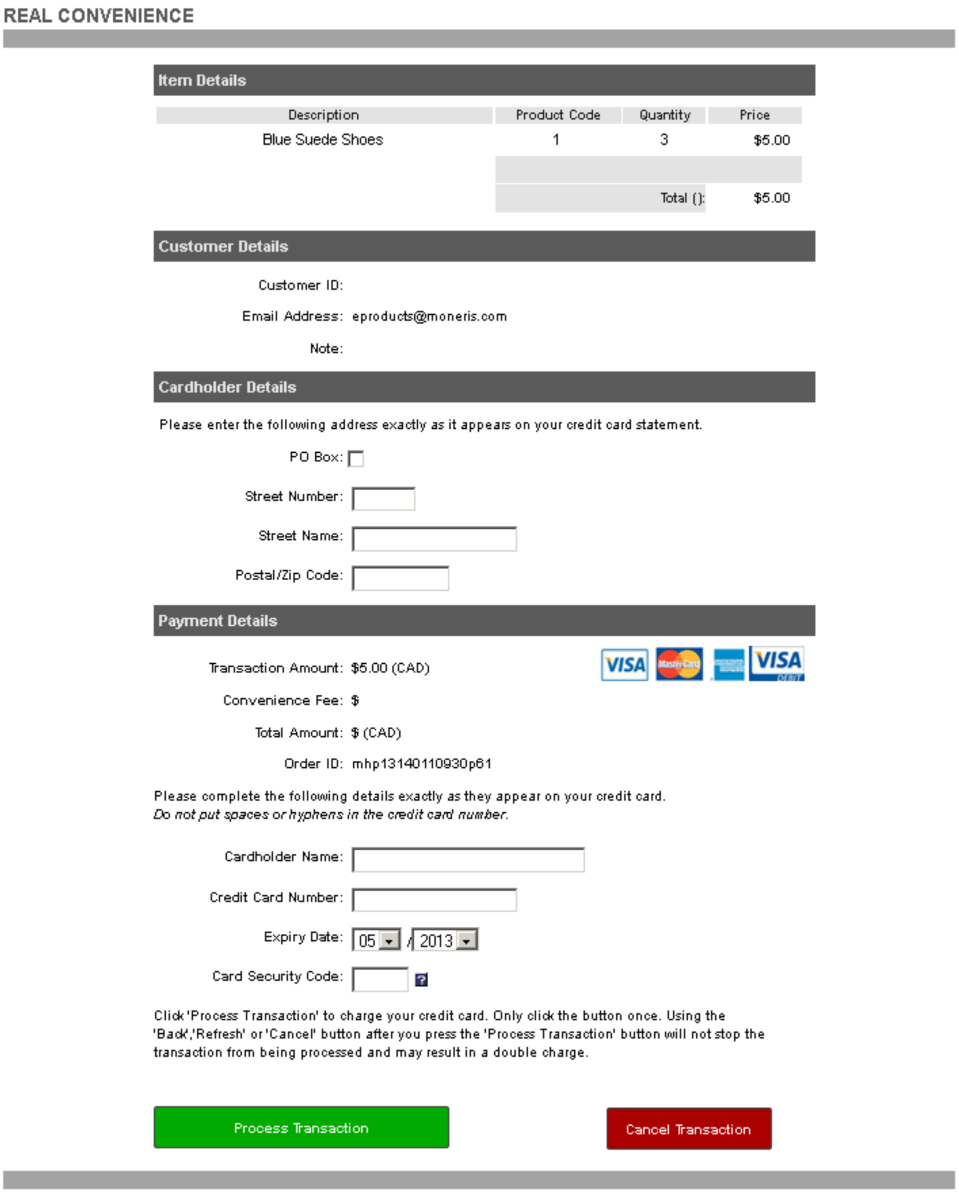
Please complete the following details exactly as they appear on your credit card.
Do not put spaces or hyphens in the credit card number.

Cardholder Name:
Credit Card Number:
Expiry Date: 06 / 2013
Card Security Code:

Click 'Process Transaction' to charge your credit card. Only click the button once. Using the 'Back','Refresh' or 'Cancel' button after you press the 'Process Transaction' button will not stop the transaction from being processed and may result in a double charge.

Appendix C Sample Hosted Paypage with Convenience Fee

NOTE: Convenience Fee calculated after card number is entered, as seen below.



REAL CONVENIENCE

Item Details

Description	Product Code	Quantity	Price
Blue Suede Shoes	1	3	$5.00
		Total ():	$5.00

Customer Details

Customer ID:

Email Address: eproducts@moneris.com

Note:

Cardholder Details

Please enter the following address exactly as it appears on your credit card statement.

PO Box: □

Street Number:

Street Name:

Postal/Zip Code:

Payment Details

Transaction Amount: $5.00 (CAD)

Convenience Fee: $

Total Amount: $ (CAD)

Order ID: mhp13140110930p61

Please complete the following details exactly as they appear on your credit card.
Do not put spaces or hyphens in the credit card number.

Cardholder Name:

Credit Card Number:

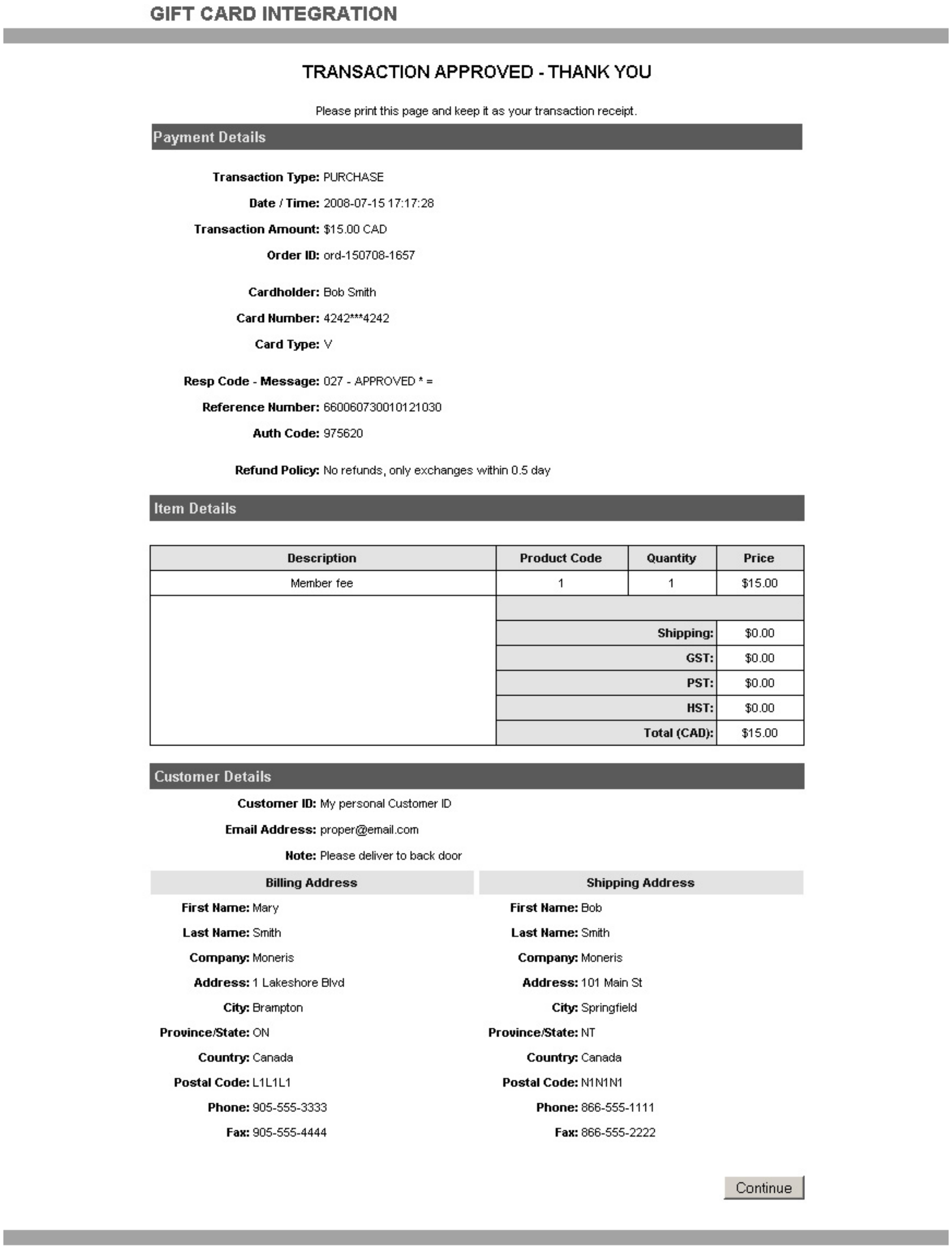
Expiry Date: 05 / 2013

Card Security Code:

Click 'Process Transaction' to charge your credit card. Only click the button once. Using the 'Back','Refresh' or 'Cancel' button after you press the 'Process Transaction' button will not stop the transaction from being processed and may result in a double charge.

Appendix D Sample Receipt

The sample below displays the layout of the response receipt. This receipt is only displayed if the Response Method is set to **Moneris Gateway will generate a receipt** in the Hosted Payment Page configuration. The sections and subsections displayed, as well as the colour and style of the receipt, are determined by the settings chosen in the Hosted Payment Page configuration. Please refer to "Configuring the Hosted Payment Page" (page 15) for further information.



GIFT CARD INTEGRATION

TRANSACTION APPROVED - THANK YOU

Please print this page and keep it as your transaction receipt.

Payment Details

Transaction Type: PURCHASE
Date / Time: 2008-07-15 17:17:28
Transaction Amount: $15.00 CAD
Order ID: ord-150708-1657

Cardholder: Bob Smith
Card Number: 4242***4242
Card Type: V

Resp Code - Message: 027 - APPROVED * =
Reference Number: 660060730010121030
Auth Code: 975620

Refund Policy: No refunds, only exchanges within 0.5 day

Item Details

Description	Product Code	Quantity	Price
Member fee	1	1	$15.00
	Shipping:		$0.00
	GST:		$0.00
	PST:		$0.00
	HST:		$0.00
	Total (CAD):		$15.00

Customer Details

Customer ID: My personal Customer ID
Email Address: proper@email.com
Note: Please deliver to back door

Billing Address		Shipping Address	
First Name:	Mary	**First Name:**	Bob
Last Name:	Smith	**Last Name:**	Smith
Company:	Moneris	**Company:**	Moneris
Address:	1 Lakeshore Blvd	**Address:**	101 Main St
City:	Brampton	**City:**	Springfield
Province/State:	ON	**Province/State:**	NT
Country:	Canada	**Country:**	Canada
Postal Code:	L1L1L1	**Postal Code:**	N1N1N1
Phone:	905-555-3333	**Phone:**	866-555-1111
Fax:	905-555-4444	**Fax:**	866-555-2222

Appendix E XML POST Response for Financial Transaction

The XML is returned in a field called xml_response. A sample of the XML is below. Fields in blue are returned optionally or based on the transaction type performed. The XML should be parsed dynamically to ensure that if and when fields are added in the future the transaction responses are handled properly. Field definitions are the same as indicated in the standard response format tables. Gift card response fields are defined below. The gift_card tag may appear once or twice depending on the number of gift cards used during the transaction. The item tag will appear for every item that was posted in the request with a quantity greater than 0.

NOTE: The XML may not be returned formatted in the manner below. It may be returned as a single line or with line breaks. Your XML parser should be able to handle these variations.

```
<?xml version='1.0' standalone='yes'?>
<response>
<response_order_id>mhp1573006623</response_order_id>
<bank_transaction_id>660035520011120030</bank_transaction_id>
<response_code>027</response_code>
<iso_code>01</iso_code>
<bank_approval_code>608681</bank_approval_code>
<time_stamp>18:53:27</time_stamp>
<date_stamp>2008-07-10</date_stamp>
<trans_name>purchase</trans_name>
<message>APPROVED * =</message>
<charge_total>1.00</charge_total>
<cardholder>Bill Smith</cardholder>
<card_num>4510***5010</card_num>
<card>V</card>
<expiry_date>0807</expiry_date>
<result>1</result>
<convenience_fee>1.00</convenience_fee>
<eci>7</eci>
<txn_num>829-0_22</txn_num>
<rvar1>1_rvar</rvar1>
<rvar2>2_rvar</rvar2>
<transactionKey>uJv2RGGasX4Kd3Tlz3eujRAY5wUCd1</transactionKey>
<recur_result> </recur_result>
<gift_card>
<order_no>mhp1573006623_g1</order_no>
<txn_num>9041-1215730102475-00035540_21</txn_num>
<response_code>000</response_code>
<ref_num>37286815</ref_num>
<terminal_id>00035540</terminal_id>
<txn_type>purchase</txn_type>
<card_num>0211***0222</card_num>
<card_desc>Gift Fixed Reloadabl</card_desc>
<date_time>Jul 10 2008 06:53PM</date_time>
<gift_charge_total>99.50</gift_charge_total>
<rem_balance>0.00</rem_balance>
<display_text>Approved</display_text>
<receipt_text>En
NHLJ: Jul 10, 2008
NHC: 07/10/2008
NHRJ: 10-07-2008
DHLJ: 10 Jul 2008D
```

```
DHC: Jul 10, 2008
DHRJ: End Of Text
</receipt_text>
<voucher_text> </voucher_text>
<result>1</result>
</gift_card>
<item>
<quantity>3</quantity>
<description>qunat 3</description>
<id>sku123</id>
<price>4.00</price>
</item>
<item>
<quantity>2</quantity>
<description>qunat 2</description>
<id>2sku123</id>
<price>24.00</price>
</item>
<item_misc>
<shipping_cost>4.03</shipping_cost>
<hst></hst>
<pst></pst>
<gst>3.03</gst>
</item_misc>
<shipping>
<ship_first_name>sfn</ship_first_name>
<ship_last_name>sln</ship_last_name>
<ship_company_name>scn</ship_company_name>
<ship_address_one>sao</ship_address_one>
<ship_state_or_province>ssop</ship_state_or_province>
<ship_postal_code>spc</ship_postal_code>
<ship_country>scount</ship_country>
<ship_phone>sp</ship_phone>
<ship_fax>sf</ship_fax>
</shipping>
<billing>
<bill_first_name>bfn</bill_first_name>
<bill_last_name>bln</bill_last_name>
<bill_company_name>bcn</bill_company_name>
<bill_address_one>bao</bill_address_one>
<bill_state_or_province>bsop</bill_state_or_province>
<bill_postal_code>bpc</bill_postal_code>
<bill_country>bcount</bill_country>
<bill_phone></bill_phone>
<bill_fax></bill_fax>
</billing>
<od_other>
<email>bill.smith@example.com</email>
<cust_id>customer num</cust_id>
<note>these are special instructions</note>
</od_other>
</response>
```

Appendix F Internet Explorer 7 Compatibility

The example code in 3.2.2.1 Getting a Temporary Token on page 57 does not support Internet Explorer 7, so if compatibility with Internet Explorer is required then the below code should be used instead.

The example below will work once you add your Hosted Tokenization ID along with your own URL.

```
<html>
<head>
<script type="text/javascript">
function doMonerisSubmit()
{
var frame_ref;

if(navigator.userAgent.indexOf("Safari") != -1)
{
frame_ref = frames["monerisFrame"];
}
else
{
frame_ref = document.getElementById("monerisFrame").contentWindow;
}
frame_ref.location = "https://esqa.moneris.com/HPPtoken/index.php?id=<YOUR HPP TOKEN
    ID>&poll=true&css_body=background:green;border:2px dotted purple;&css_textbox=border:1px solid
    blue; &css_textbox_pan=width:140px;&enable_exp=1&css_textbox_exp=width:40px;&enable_cvd=1&css_
    textbox_cvd=width:40px;&parent=<THE URL OF PARENT WINDOW> " + "#submitResForm" + (new Date
    ()).getTime();
}
var post_data = "";

function checkForMessages()
{
if(location.hash != post_data)
{
post_data = location.hash;
var raw_json = decodeURIComponent(post_data.substr(1));
var respData = eval("(" + raw_json + ")");
document.getElementById("monerisResponse").innerHTML = " SENT " + " - " +
respData.responseCode + " " + respData.dataKey + " - bin: " + respData.bin;
document.getElementById("monerisFrame").style.display = 'none';
}
}
setInterval("checkForMessages()", 200);
</script>
</head>
<body style=background:#E3E3E3>
<div>This is the outer page!!</div>
<div id=monerisResponse></div>
<iframe id=monerisFrame src="https://esqa.moneris.com/HPPtoken/index.php?id=<YOUR HPP TOKEN
    ID>&poll=true&css_body=background:green;border:2px dotted purple;&css_textbox=border:1px solid
    blue &css_textbox_pan=width:140px;&enable_exp=1&css_textbox_exp=width:40px;&enable_cvd=1&css_
    textbox_cvd=width:40px;&parent=<THE URL OF PARENT WINDOW>" frameborder='0' width="200px"
    height="30px"></iframe>
<input type=button onClick=doMonerisSubmit() value="go go">
</html>
```

Appendix G How to Identify Visa Debit Cards

Visa Debit transactions can be indentified by checking the value of the visaDebit member of the response object.

The example below demonstrates how to do this in bold.

NOTE: the response value for this field will be either True or undefined, the field will not return false, so the best way to handle this is to check if the response is equal to the string value of “true”.

```
<html>
<head>
<title> Outer Frame - Merchant Page</title>
<script>
    function doMonerisSubmit()
    {
        var monFrameRef = document.getElementById('monerisFrame').contentWindow;
        monFrameRef.postMessage('','https://esqa.moneris.com/HPPtoken/index.php');
        return false;
    }

    var respMsg = function(e)
    {
        var respData = eval("(" + e.data + ")");

        document.getElementById("monerisResponse").innerHTML = e.origin + " SENT " + " | Response
    Code: "
+ respData.responseCode + " | "Temporary Token: " + respData.dataKey + " | Error Message: "
+ respData.errorMessage + " | Visa Debit: " + respData.visaDebit;
        document.getElementById("monerisFrame").style.display = 'none';
        //Correct what to check for a Visa Debit card
        if(respData.visaDebit == "true")
        {
            alert("This is a Visa Debit Card");
        }
    }

    window.onload = function()
    {
        if (window.addEventListener)
        {
            window.addEventListener ("message", respMsg, false);
        }
        else
        {
            if (window.attachEvent)
            {
                window.attachEvent("onmessage", respMsg);
            }
        }
    }
</script>
</head>

<body>
    <div>This is the outer page!!</div>
    <div id=monerisResponse></div>
    <iframe id=monerisFrame src="https://esqa.moneris.com/HPPtoken/index.php?id=<YOUR HPP TOKEN
    ID>&poll=true&css_body=background:green;border:2px dotted purple;&css_textbox=border:1px solid
    blue &css_textbox_pan=width:140px;&enable_exp=1&css_textbox_exp=width:40px;&enable_cvd=1&css_
    textbox_cvd=width:40px;&parent=<THE URL OF PARENT WINDOW>" frameborder='0' width="200px"
    height="30px"></iframe>

    <input type=button onClick=doMonerisSubmit() value="go go">

</html>
```