Merchant Integration Guide - Hosted Payment Page

United States

Hosted Payment Page

Convenience Fee

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: 1-866-696-0488 (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 295) 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.

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 72.
5. **Moving the solution into production**. For more information on this, see 5 Moving to Production on page 90

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 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.

1.4 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.

1.5 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.

1.6 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.

1.7 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 102

1.8 MasterPass

MasterPass is a digital wallet service offered to MasterCard cardholders.

1.9 Visa Checkout

Visa Checkout is a digital wallet service offered to Visa cardholders.

2 Hosted Payment Page Configuration Tool

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://esplusqa.moneris.com/usmpg

and use one of the following login IDs.

Table 1: Test IDs for Merchant Resource Center

Store ID	Username	Password
monusqa002[1]	DemoUser	abc1234
monusqa003	DemoUser	abc1234
monusqa004	DemoUser	abc1234
monusqa005	DemoUser	abc1234
monusqa024[2]	DemoUser	abc1234
monusqa025[3]	DemoUser	abc1234
monusqa138[4]	DemoUser	abc1234

[1] test store 'monusqa002' is intended for testing Gift and Loyalty transactions

[2] test store 'monusqsa024' is intended for testing ACH transactions only

[3] test store 'monusqa025' is intended for testing both ACH and Credit Card transactions

[4] test store 'monusqa138' is intended for testing Convenience Fee transactions

Table 2: Test IDs for Merchant Resource Center - Convenience Fee

Store ID	Username	Password
monca00392	DemoUser	abc1234

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://esplusqa.moneris.com/usmpg
   - Production: https://esplus.moneris.com/usmpg
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 (hpp_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://esplusqa.moneris.com/usmpg
   - Production: https://esplus.moneris.com/usmpg
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://esplusqa.moneris.com/usmpg
   - Production: https://esplus.moneris.com/usmpg
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 "hpp_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 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 Gift Cards, the transaction type that will be processed will be the one defined in the default 'Transaction Type' section above.

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.

Colors and Styles

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

Background Color

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 Color

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

Company Name Color

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

Header and Footer Highlight Color

This defines the color 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 Color

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

Section Divider Text Color

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

Subsection Divider Color

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

Subsection Divider Text Color

Occasionally the Subsection Divider will contain text — this defines the color 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, client_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 (crypt_type) 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 crypt_type value sent in a follow-on transaction request must reflect the crypt_type value received from the Preauthorization.

EXAMPLE: if the crypt_type value received from the preauthorization is "5", then the subsequent API capture request must also have its crypt_type 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_no 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 = é) 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://esplusqa.moneris.com/usmpg which will display the URL you are posting from.

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 https://esplus.moneris.com/usmpg 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_no 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 Color Chart

All colours in the Colors 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 Color

This defines the background color 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 Color

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

Company Name Color

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

Header & Footer Highlight Color

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

Section Divider Color

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

Section Divider Text Color

Occasionally the Section Divider Bar will contain text – this defines the color 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 color for the Section Divider Color and for the Section Divider Text Color).

Subsection Divider Color

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

Subsection Divider Text Color

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

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:

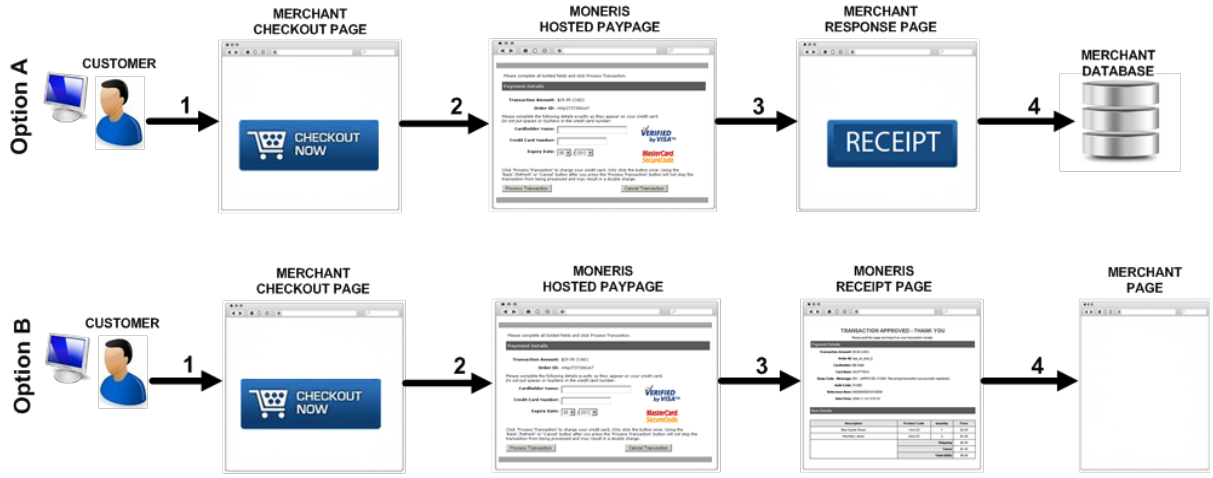
DO NOT USE REAL ACCOUNT INFORMATION WHEN TESTING IN THE QA ENVIRONMENT.

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 43) 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 ACH banking details 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 or ACH banking details 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

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 46)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://esplusqa.moneris.com/HPPDP/index.php - Development https://esplus.moneris.com/HPPDP/index.php - Production
hpp_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.
amount	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://esplusqa.moneris.com/DPHPP/index.php" >
<INPUT TYPE="HIDDEN" NAME="hpp_id" VALUE="QRZX9qa002">
<INPUT TYPE="HIDDEN" NAME="hpp_key" VALUE="hpTTJYGTDLNB">
<INPUT TYPE="HIDDEN" NAME="amount" 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
li_idn	hidden	Product Code - SKU (max 10 chars)
li_descriptionn	hidden	Product Description - (max 15 chars)
li_quantityn	hidden	Quantity of Goods Purchased - (max - 4 digits)
li_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)
li_taxes	hidden	Total tax amount - (min 0.01 & max 9999999.99)

NOTE: You must send a li_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***, li_descriptionn, li_quantityn,li_pricen and subtotal***n*** details for the item to be displayed in the receipt.

Sample - Basic Transaction with Item Details

```
<input type="hidden" name="li_id1" value="1">
<input type="hidden" name="li_description1" value="Blue Suede Shoes">
<input type="hidden" name="li_quantity1" value="3">
<input type="hidden" name="li_price1" value="40.00">
<input type="hidden" name="li_id2" value="2">
<input type="hidden" name="li_description2" value="Red Leather Shoes">
<input type="hidden" name="li_quantity2" value="3">
<input type="hidden" name="li_price2" value="80.00">
<input type="hidden" name="li_shipping" value="14.00">
<input type="hidden" name="li_taxes" value="14.00">
```

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_no	hidden	MUST be unique per transaction and be no more than 50 characters. System will generate a value if excluded.
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.
client_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.

Variable name	Type	Description
crypt_type	hidden	If using the Hosted Payment Page to integrate an internal order management system for Mail/Telephone Orders, send an crypt_typevalue of 1. If the Hosted Pay Page supports VbV/MCSC, the crypt_type generated by VbV/MCSC will override the value passed in.

NOTE:

The order_no 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_no, and the customer's e-mail and special instructions, if any, 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_no" VALUE ="oid43333">
<INPUT TYPE="TEXT" NAME="client_email" VALUE="john.smith@moneris.com">
<INPUT TYPE="TEXT" NAME="note" VALUE="All deliveries go to back door">
```

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
od_ship_firstname	text
od_ship_lastname	text
od_ship_company	text
od_ship_address	text
od_ship_city	text
od_ship_state	text
od_ship_zipcode	text
od_ship_country	text
od_ship_phone	text
od_ship_fax	text
od_bill_firstname	text
od_bill_lastname	text
od_bill_company	text
od_bill_address	text
od_bill_city	text
od_bill_state	text
od_bill_zipcode	text
od_bill_country	text
od_bill_phone	text
od_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="od_bill_firstname" VALUE="John">
<INPUT TYPE="HIDDEN" NAME="od_bill_lastname" VALUE="Smith">
<INPUT TYPE="HIDDEN" NAME="od_bill_company" VALUE="Moneris Solutions">
<INPUT TYPE="HIDDEN" NAME="od_bill_address" VALUE="101 Main St">
<INPUT TYPE="HIDDEN" NAME="od_bill_city" VALUE="Smallville">
<INPUT TYPE="HIDDEN" NAME="od_bill_state" VALUE="NY">
<INPUT TYPE="HIDDEN" NAME="od_bill_zipcode" VALUE="123456">
<INPUT TYPE="HIDDEN" NAME="od_bill_country" VALUE="USA">
<INPUT TYPE="HIDDEN" NAME="od_bill_phone" VALUE="555-555-5555">
<INPUT TYPE="HIDDENV NAME="od_bill_fax" VALUE="555-555-6666">
<INPUT TYPE="HIDDEN" NAME="od_ship_firstname" VALUE="Jen">
```

Sample Set Billing and Shipping Details - CA

```
<INPUT TYPE="HIDDEN" NAME="od_ship_lastname" VALUE="Smith">
<INPUT TYPE="HIDDEN" NAME="od_ship_company" VALUE="Moneris Solutions">
<INPUT TYPE="HIDDEN" NAME="od_ship_address" VALUE="150 Lakeshore Rd">
<INPUT TYPE="HIDDEN" NAME="od_ship_city" VALUE="Springfield">
<INPUT TYPE="HIDDEN" NAME="od_ship_state" VALUE="IL">
<INPUT TYPE="HIDDEN" NAME="od_ship_zipcode" VALUE="234567">
<INPUT TYPE="HIDDEN" NAME="od_ship_country" VALUE="USA">
<INPUT TYPE="HIDDEN" NAME="od_ship_phone" VALUE="333-555-5555">
<INPUT TYPE="HIDDEN" NAME="od_ship_fax" VALUE="333-555-6666">
```

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 84) 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,

Variable name	Type	Description
		rvarMyVariable).

Sample Optional 'rvar' Request

```
<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">
```

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 1).

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
recur_initiate	hidden	1 = initiate a Recurring transaction, anything else will not initiate a recurring transaction
recur_unit	hidden	Must be "day", "week" "month" or "eom" (end of month). This is the base unit for the recurring interval.
recur_period	hidden	Numeric value. The period is used in conjunction with recur_ unit to determine the interval between payments. **EXAMPLE:** recur_unit = "month" and recur_period = "12" the charge will be billed once every twelve months. If recur_unit= "week" and recur_period = "1" the charge will be billed once a week. If recur_unit = "eom" and recur_period = "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.

Variable name	Type	Description
recur_start_date	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 recur_start_now to bill card holder immediately).
recur_start_now	hidden	"true" / "false" This will charge the transaction immediately and then initiate recurring billing to commence on the recur_start_date. **EXAMPLE:** To charge a card immediately (assuming the date is May 9, 2006) and then bill every month thereafter: recur_start_now = "true" recurStartDate = "2006/05/09" – set the date 1 month after the present recur_unit = "month" recur_period = "1"
recur_amount	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 recur_amount 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

Variable name	Type	Description
		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. amount = "20.00" recur_amount = "40.00" recur_start_now = "true" recur_start_date = "2006/06/01" recur_unit = "month" recur_period =”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 recur_unit and recur_period settings over a 5 year period. **EXAMPLE:** If you would like to bill a client indefinitely you should set the recur_ num 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 recur_num = "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, hpp_id and amount.

This transaction will bill the amount sent in amount immediately (using the example above, amount="1.00") Then commencing June 1, 2007 (assuming that this is a date in the future) the card will be billed $4.00 every 2 weeks, 26 times (one year).

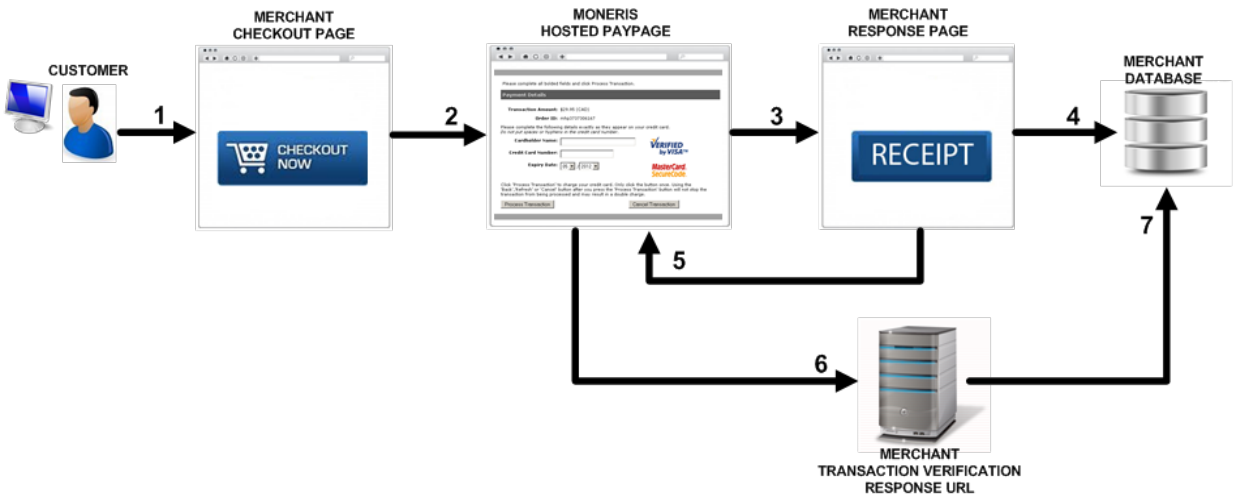
Sample Initiate Recurring Transaction

```
<input type="hidden" name="recur_initiate" value="1">
<input type="hidden" name="recur_unit" value="week">
<input type="hidden" name="recur_period" value="2">
<input type="hidden" name="recur_num" value="26">
<input type="hidden" name="recur_start_now" value="true">
<input type="hidden" name="recur_amount" value="4.00">
<input type="hidden" name="recur_start_date" value="2007/06/01">
```

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 43) 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 43).
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://esplusqa.moneris.com/HPPDP/index.php Production: https://esplus.moneris.com/HPPDP/index.php
hpp_id	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool
hpp_key	hidden	Provided by Moneris Solutions – Hosted Payment Page Configuration Tool
verify_key	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 ACTION="https://esplusqa.moneris.com/DPHPP/index.php" method="POST">
<INPUT TYPE="hidden" NAME="hpp_id" VALUE="QRZX9qa002">
<INPUT TYPE="hidden" NAME="hpp_key" VALUE="hpTTJYGTDLNB">
<INPUT TYPE="hidden" NAME="verify_key"
    VALUE="%C0%1F%F7hE%11x5%10%255%93j%D2%89%A9ID%7D3%97s%BE%DF%C6%C0">
<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
order_no	50-character alphanumeric	order_id of the original transaction
response_code	3-character alphanumeric	Transaction Response Code from the original transaction

Variable name	Type	Description
		< 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.
verify_key	100-character alphanumeric	The verify_key from the request
message	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

Variable name	Type	Description
		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.

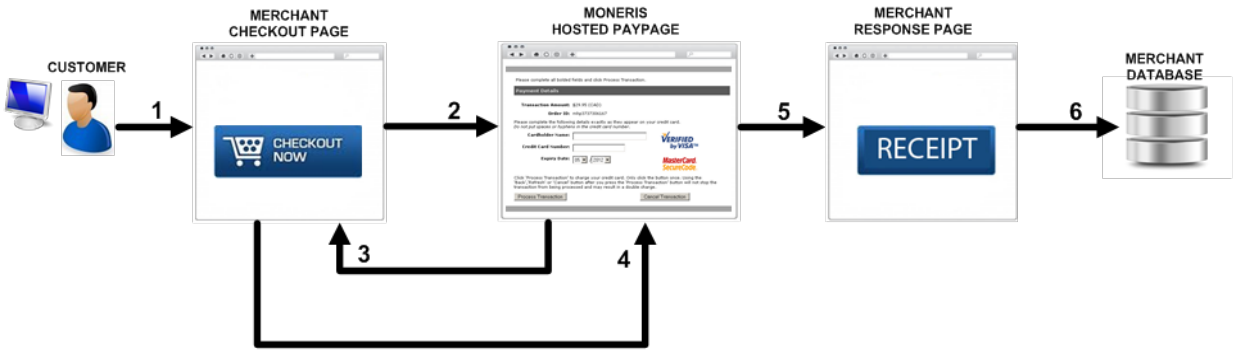
Below is a sample of a valid Transaction Verification Response displayed on our server in XML format:

Sample Transaction Verification Response (Valid)	Sample Transaction Verification Response (Invalid)
<?xml version="1.0" standalone="yes"?> <response> <order_no>dp_cr_test_tr13</order_no> <response_code>1</response_code> <amount>10.24</amount> <txn_num>3270-0_1</txn_num> <message>Valid - Approved</message > </response>	<?xml version="1.0" standalone="yes"?> <response> <order_no>dp_mr_test_100005</order_no> <response_code>995</response_code> <amount>null</amount> <txn_num>1247-0_1</txn_num> <message>Invalid</message > </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 47) 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
hpp_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

Variable name	Description
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://esplusqa.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>
```

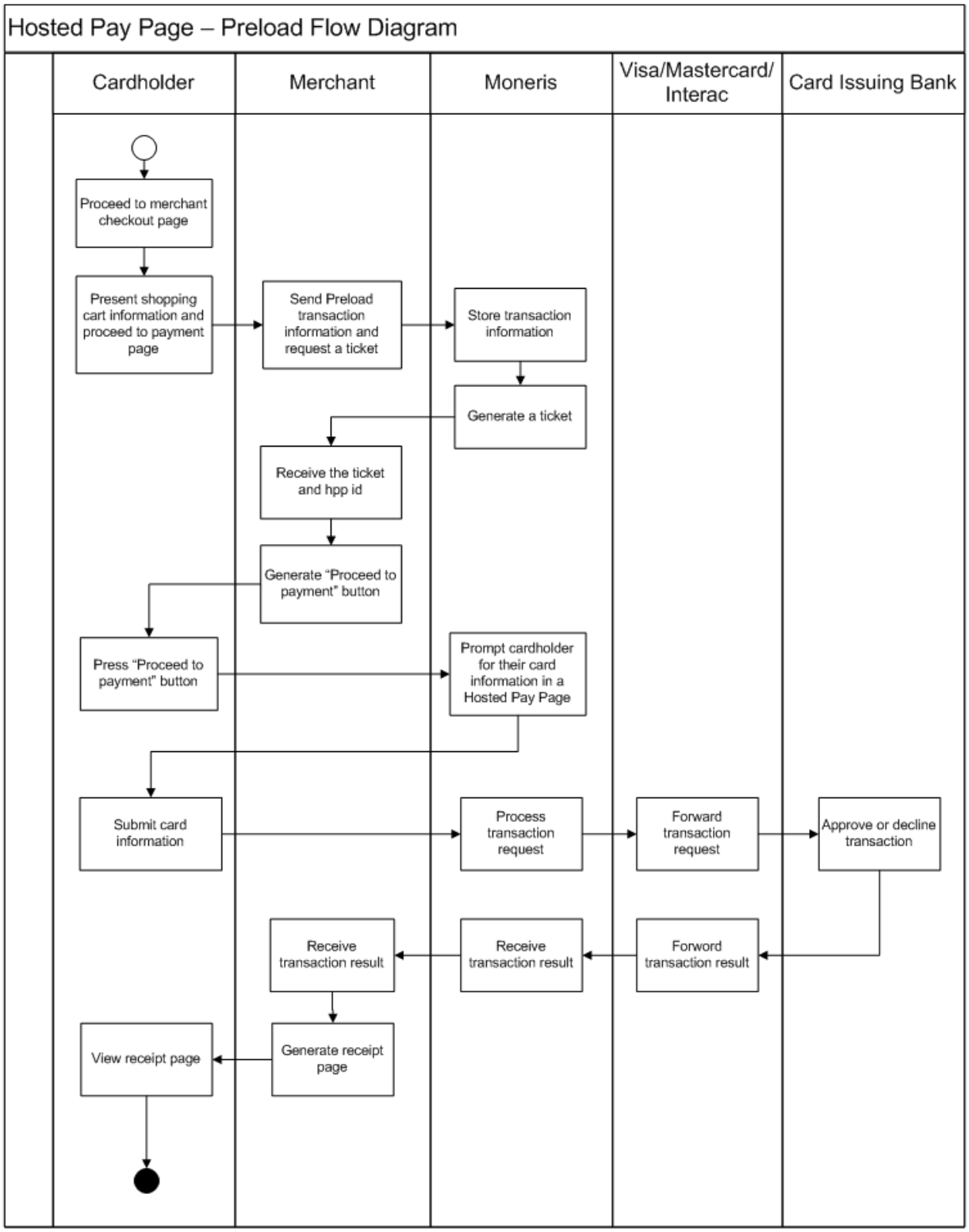
Step 4 - Normal Hosted Payment Page Process

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.

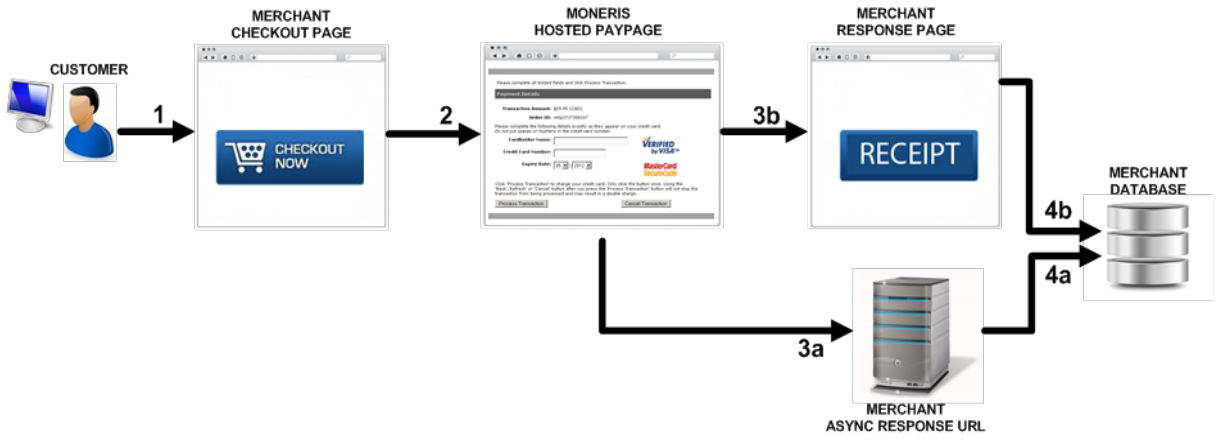
Data Preload Diagram



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=
<?xml version='1.0' standalone='yes'?>
<response>
<order_no>mhp4039836065</order_no>
<ref_num>640000030014662620</ref_num>
<response_code>000</response_code>
<auth_code>898387</auth_code>
<txn_time>15:53:47</txn_time>
<txn_date>2012-06-22</txn_date>
<txn_type>purchase</txn_type>
<message>APPROVED*</message>
<amount>10.00</amount>
<total_amount></total_amount>
<cardholder>Bob Smith</cardholder>
<card_num>3735***5005</card_num>
<card_type>AX</card_type>
<exp_month>06</exp_month>
<exp_year>12</exp_year>
<txn_num>608634-0_10</txn_num>
<crypt_type>7</crypt_type>
<result>1</result>
<rvar_1>1</rvar_1>
<rvar_monkey>monkeys are funny</rvar_monkey>
```

Sample Asynchronous Transaction Response - XML

```
<rvar_123abc>123abc</rvar_123abc>
<avs_response_code> </avs_response_code>
<cvd_response_code>N</cvd_response_code>
</response>
```

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

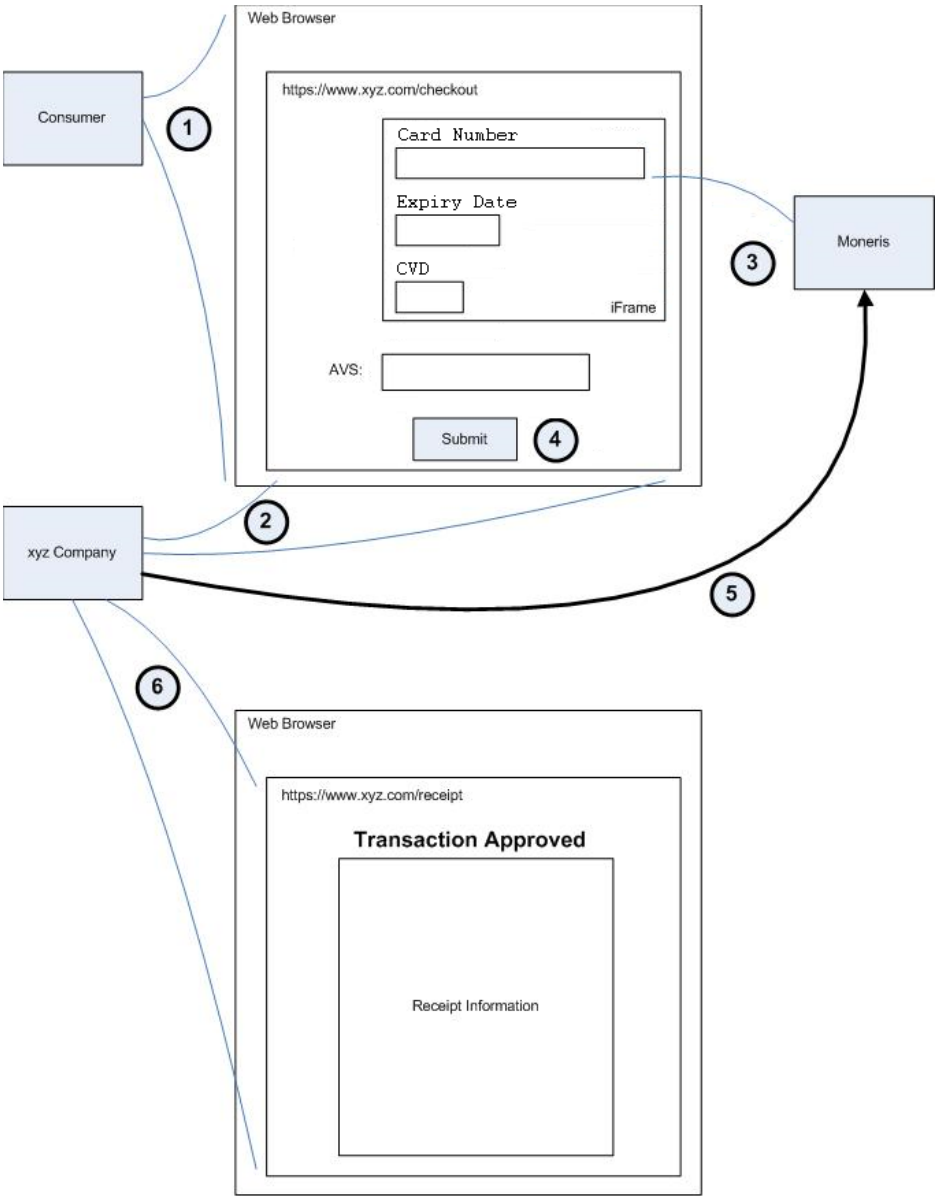
Sample Script to Read Asynchronous Transaction Response

```
<?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 .= "Order ID: " . $receipt["order_no"] . "\r\n";
$results .= "Reference Number: " . $receipt["ref_num"] . "\r\n";
$results .= "Response Code: " . $receipt["response_code"] . "\r\n";
$results .= "Auth Code: " . $receipt["auth_code"] . "\r\n";
$results .= "Transaction Time: " . $receipt["txn_time"] . "\r\n";
$results .= "Transaction Date: " . $receipt["txn_date"] . "\r\n";
$results .= "Transaction Type: " . $receipt["txn_type"] . "\r\n";
$results .= "Message: " . $receipt["message"] . "\r\n";
$results .= "Amount: " . $receipt["amount"] . "\r\n";
$results .= "Cardholder Name: " . $receipt["cardholder"] . "\r\n";
$results .= "Card Number: " . $receipt["card_num"] . "\r\n";
$results .= "Card Type: " . $receipt["card_type"] . "\r\n";
$results .= "Expiry Date(YYMM): " .$receipt["exp_year"].$receipt["exp_month"]."\r\n";
$results .= "Transaction ID: " . $receipt["txn_num"] . "\r\n";
$results .= "ECI: " . $receipt["crypt_type"] . "\r\n";
$results .= "Result: " . $receipt["result"] . "\r\n";
$results .= "AVS Result Code: " . $receipt["avs_response_code"] . "\r\n";
$results .= "CVD Result Code: " . $receipt["cvd_response_code"] . "\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



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

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 I 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;
```

Sample Getting 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';
}
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.

Variable name	Definition
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
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)

Code	Message/Description
941	Error generating token
942	Invalid Profile ID, or source URL
943	Card data is invalid (not numeric, fails mod10, we will remove spaces)
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.
```

Sample Processing the Payment - Hosted Tokenization

```
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);
// 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

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

Table 2: Required Variables for Adding Vault Profile - ACH

Variable name	Type	Description
ach_sec	hidden	SEC code – possible values are : web – Internet Initiated Entry ccd – Cash Concentration or Disbursement ppd – Prearranged Payment and Deposit Your Moneris Gateway merchant account must be properly configured to accept each of the SEC code types that are used.

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://esplusqa.moneris.com/DPHPP/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">
<INPUT TYPE="HIDDEN" NAME="ach_sec" VALUE="web">
<INPUT TYPE="HIDDEN" NAME="pd_p_account_number" VALUE="1234567">
<INPUT TYPE="HIDDEN" NAME="pd_presentation_type" VALUE="W">
<!--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
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

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

Table 2: Optional Variables - Adding New Profile to Vault - ACH Transactions

Variable name	Type	Description
ach_cust_first_name	50-character alphanumeric	Client's first name
ach_cust_last_name	50-character alphanumeric	Client's last name
ach_cust_address1	50-character alphanumeric	Client's address
ach_cust_address2	50-character alphanumeric	Client's extra address information
ach_cust_city	50-character alphanumeric	Client's city
ach_cust_state	20-character alphanumeric	Client's state
ach_cust_zip	10-character alphanumeric	Client's ZIP code

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">
```

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://esplusqa.moneris.com/HPPDP/index.php Production: https://esplus.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 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 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 ACTION="https://esplusqa.moneris.com/DPHPP/index.php" method="POST">
<INPUT TYPE="hidden" NAME="res_id" VALUE="QRZX9qa002">
<INPUT TYPE="hidden" NAME="res_key" VALUE="resTTJYGTDLNB">
<INPUT TYPE="hidden" NAME="verify_key" VALUE="SADF98AF78ADSFUASDF987">
<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 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> <msg>Valid - Registered</msg > </response>	<?xml version="1.0" standalone="yes"?> <response> <response_code>995</response_code> <message>Invalid</message > </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 or 4005554444444403
Amex	373599005095005
Diners	36462462742008

When testing ACH transactions you may use the following test bank account details:

Table 2: Test Account Details - ACH

Financial Institution	Routing Number	Account Number	Check Number
FEDERAL RESERVE BANK	011000015	Any number between 5-22 digits	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 90) 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
order_no	50-character alphanumeric	order_no 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
txn_date	yyyy-mm-dd	Processing host date stamp
txn_time	##:##:##	Processing host time stamp
auth_code	8-character alphanumeric	Authorization code returned from the issuing institution
result	1-character numeric	1 = approved , 0 = declined, incomplete
txn_type	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).

Variable name	Size/Type	Description
		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". achdebit: bank account information is collected and funds are debited from this account 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.
cardholder	40-character alphanumeric	Cardholder's name
amount	9-character decimal. Up to 7-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 1234567.89	Amount of the transaction
card_type	2-character alphanumeric	Credit Card Type M = Mastercard V = Visa AX = American Express DC = Diners Card NO = Novus / Discover C = JCB SE = Sears CQ = ACH (Online Check)

Variable name	Size/Type	Description
card_num	####***#### ***####	First 4 and last 4 digits of the card # Last 4 digits of the bank account number (ACH/Online Check)
exp_month	2-character numeric	2 digit month (ex. 01, 02...). Will return expiry month entered on the Hosted Payment Page (Credit Card only)
exp_year	2-character numeric	2 digit year (ex. 01, 02...). Will return expiry year entered on the Hosted Payment Page (Credit Card only)
message	100-character alphanumeric	Response description returned from issuing institution or from Moneris Gateway if there is a system error.
CfStatus	2-character alphanumeric	Indicates the status of the merchant and convenience fee transactions. The CfStatus field provides details about the transaction behavior and should be referenced when contacting Moneris Customer Support. Possible values are: 1 or 1F = Completed 1st purchase transaction 2 or 2F = Completed 2nd purchase transaction 3 = Completed void transaction 4A or 4D = Completed refund transaction 7 or 7F = Completed merchant independent refund transaction 8 or 8F = Completed merchant refund transaction 9 or 9F = Completed 1st void

Variable name	Size/Type	Description
		transaction 10 or 10F = Completed 2nd void transaction 11A or 11D = Completed refund transaction
ref_num	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. **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.
verify_key	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

<table>
<tr><th>Variable name</th><th>Size/Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>transaction verification functionality.</td></tr>
<tr><td>ticket</td><td>alphanumeric</td><td>The value returned from the preload data request.

NOTE: This variable applies only when using data preload functionality.</td></tr>
<tr><td>rvarn</td><td>optional</td><td>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 included in the response.</td></tr>
<tr><td>crypt_type</td><td>1-character numeric</td><td>Electronic Commerce Indicator that was sent with the transaction.

Possible values are:

<table>
<tr><th>Crypt Type</th><th>TVisa/MCSC Definitions</th></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</td></tr>
</table>
</td></tr>
</table>

Variable name	Size/Type	Description
		<table><tr><td></td><td>- 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>
txn_num	20-character alphanumeric	Gateway Transaction identifier. This value is required if merchant decides to send automated captures, voids or 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_result	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_result	1-character alphanumeric	Indicates the CVD validation

Variable name	Size/Type	Description
		result. Refer to Appendix A Transaction Request Examples for further details. 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 crypt_type returned is a “6” and the result code is a "B", it becomes liable for chargeback. Please see CAVV Result Codes - Verified by Visa (see page 87) for the CAVV result codes table

Table 2: 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

Variable name	Size/Type	Description
		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 3: Response Fields - Convenience Fee Transactions

Variable name	Size/Type	Description
FeeAmount	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 'FeeAmount' must be less than the 'amount'.
FeeRate	9-character decimal	The convenience fee rate that has been defined on the

Variable name	Size/Type	Description
		merchant's profile. For example: 1.00 – a fixed amount or 10.0 - a percentage amount
FeeType	AMT / PCT	The type of convenience fee that has been defined on the merchant's profile. Available options are: AMT – fixed amount PCT – percentage
CfSuccess	true/false	Indicates whether the Convenience Fee transaction processed successfully.

Table 4: Response Fields - Transaction Risk Management Tool Transactions

Variable name	Size/Type	Description

Table 5: 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	Amount processed on this loyalty card transaction. This value will be displayed in the

Variable name	Type	Description
	EXAMPLE: 1234567.89	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 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 6: 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.

4.3 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 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

4.3.5.1 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>
```

4.3.5.2 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
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.
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)

Result Code	Message	What this means to you as a merchant…
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)
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)

Result Code	Message	What this means to you as a merchant…
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.

5 Moving to 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://esplus.moneris.com/usmpg/activate 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://esplusqa.moneris.com/HPPDP/index.php> to contain the production URL: <FORM METHOD="POST" ACTION=https://esplus.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://esplus.moneris.com/usmpg. You can use the store administrator ID you created during the activation process and then create additional users as needed.

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

5.3 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)

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://esplusqa.moneris.com/DPHPP/index.php" METHOD="POST">
<!-- Mandatory Fields -->
<INPUT TYPE="hidden" NAME="hpp_id" VALUE="QRZX9qa002">
<INPUT TYPE="hidden" NAME="hpp_key" VALUE="hpTTJYGTDLNB">
<INPUT TYPE="hidden" NAME="amount" VALUE="9.00">
<!-- Unique Order ID -->
<INPUT TYPE="hidden" NAME="order_no" VALUE="hpp_mr_test_1">
<!-- Additional Optional Details -->
<INPUT TYPE="hidden" NAME="client_email" VALUE="john.smith@moneris.com">
<INPUT TYPE="hidden" NAME="note" VALUE="This is a note.">
<INPUT TYPE="hidden" NAME="cust_id" VALUE="Some Customer Number">

<!-- 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">
<!-- Item Information-->
<INPUT TYPE="hidden" NAME="li_quantity1" VALUE="1">
<INPUT TYPE="hidden" NAME="li_description1" VALUE="Blue Suede Shoes">
<INPUT TYPE="hidden" NAME="li_id1" VALUE="1sku123">
<INPUT TYPE="hidden" NAME="li_price1" VALUE="2.00">
<INPUT TYPE="hidden" NAME="li_quantity2" VALUE="2">
<INPUT TYPE="hidden" NAME="li_description2" VALUE="Red Mary-Janes">
<INPUT TYPE="hidden" NAME="li_id2" VALUE="2sku123">
<INPUT TYPE="hidden" NAME="li_price2" VALUE="1.00">
<INPUT TYPE="hidden" NAME="li_shipping" VALUE="4.00">
<INPUT TYPE="hidden" NAME="li_taxes" VALUE="1.00">
<!-- Billing Information -->
<INPUT TYPE="hidden" NAME="od_bill_firstname" VALUE="John">
<INPUT TYPE="hidden" NAME="od_bill_lastname" VALUE="Smith">
<INPUT TYPE="hidden" NAME="od_bill_company" VALUE="Moneris">
<INPUT TYPE="hidden" NAME="od_bill_address" VALUE="101 Main St">
<INPUT TYPE="hidden" NAME="od_bill_city" VALUE="Springfield">
<INPUT TYPE="hidden" NAME="od_bill_state" VALUE="IL">
<INPUT TYPE="hidden" NAME="od_bill_zipcode" VALUE="123456">
<INPUT TYPE="hidden" NAME="od_bill_country" VALUE="USA">
<INPUT TYPE="hidden" NAME="od_bill_phone" VALUE="555-555-5555">
<INPUT TYPE="hidden" NAME="od_bill_fax" VALUE="555-555-5566">
<!-- Shipping information -->
<INPUT TYPE="hidden" NAME="od_ship_firstname" VALUE="Mary">
<INPUT TYPE="hidden" NAME="od_ship_lastname" VALUE="Smith">
<INPUT TYPE="hidden" NAME="od_ship_company" VALUE="Moneris">
<INPUT TYPE="hidden" NAME="od_ship_address" VALUE="222 Lakeshore Blvd">
<INPUT TYPE="hidden" NAME="od_ship_city" VALUE="New York">
<INPUT TYPE="hidden" NAME="od_ship_state" VALUE="NY">
<INPUT TYPE="hidden" NAME="od_ship_zipcode" VALUE="234567">
<INPUT TYPE="hidden" NAME="od_ship_country" VALUE="USA">
<INPUT TYPE="hidden" NAME="od_ship_phone" VALUE="666-555-6666">
<INPUT TYPE="hidden" NAME="od_ship_fax" VALUE="666-555-6655">
<!-- Recurring Information -->
<INPUT TYPE="hidden" NAME="recur_initiate" VALUE="1">
<INPUT TYPE="hidden" NAME="recur_unit" VALUE="week">
<INPUT TYPE="hidden" NAME="recur_period" VALUE="1">
```

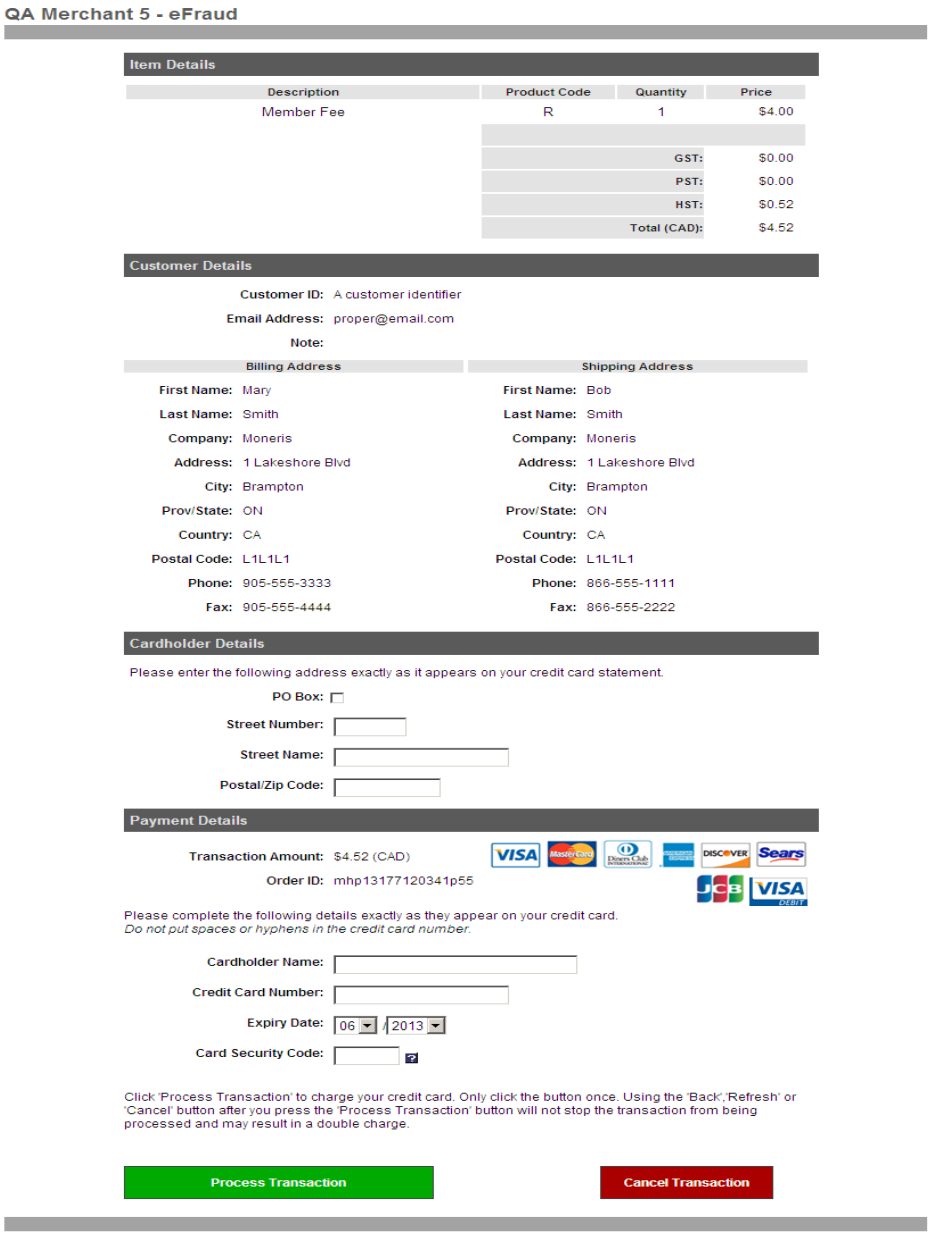
```
<INPUT TYPE="hidden" NAME="recur_num" VALUE="4">
<INPUT TYPE="hidden" NAME="recur_start_now" VALUE="true">
<INPUT TYPE="hidden" NAME="recur_amount" VALUE="3.00">
<INPUT TYPE="hidden" NAME="recur_start_date" VALUE="2006/12/01">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Click to proceed to Secure Page">
</FORM>
```

Hosted Vault Transaction Request

```
<FORM ACTION="https://esplusqa.moneris.com/DPHPP/index.php" METHOD="POST">
<!-- Mandatory Fields -->
<INPUT TYPE="hidden" NAME="res_id" VALUE="QRZX9qa002">
<INPUT TYPE="hidden" NAME="res_key" VALUE="reTTJYGTDLNB">
<!-- 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">
<!—Payment Type information -->
<INPUT TYPE="hidden" NAME="cc_crypt_type" VALUE="7">
<INPUT TYPE="hidden" NAME="ach_sec" VALUE="web">
<INPUT TYPE="hidden" NAME="pd_p_account_number" VALUE="123invoicenumber">
<INPUT TYPE="hidden" NAME="pd_presentation_type" VALUE="W">
<!-- 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 color 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.



Appendix C Credit Card Transactions

For all regular credit card transactions, the credit card associations expect certain fields to be presented to the cardholder on a receipt. These required fields are listed below, including the corresponding variable name as returned in a Hosted Payment Page GET or POST response.

	Field	Variable	Description
1	Merchant Name		The name of the store / business – this should be the same as what you have registered with Moneris Solutions.
2	Merchant URL		Web site address of the store / business.
3	Transaction Type	txn_type	The type of transaction that was performed: • Sale (Purchase) • Authorization (PreAuth) **NOTE:** NOTE: The terms listed above are the names for transactions as they are to be displayed on receipts. Other terms used for the transaction type are indicated in brackets.
4	Transaction Amount	amount	The total amount being paid by credit card.
5	Convenience Fee Amount	convenience_fee	The amount of the convenience fee. Please note, the 'Convenience Fee Amount' must be less than the 'Transaction Amount'.
6	Transaction Date and Time	txn_time txn_date	The date may be in any format, but must include the day, month and year. The time must be in 24 hour format. It is good practice to include the seconds in the time format to help with tracing transactions.

	Field	Variable	Description
7	Reference Number	ref_num	6400135 = terminal number 001 = shift 001 = batch number 001 = sequence number 0 = reserved
8	Auth Code	auth_code	The authorization number is only printed if the transaction is approved. If the transaction is declined, the title is printed but the field is blank.
9	Response Code	response_code	The 3 digit response code returned by the issuer (ex. 000 – 999)
10	Response Message	message	Message indicating whether the transaction was Approved or Declined.
11	Cardholder Name	cardholder	Display both First and Last Name as submitted by the cardholder.
12	Goods and Services Order		A list of all items/services that are being paid for in this transaction.
13	Return Policy		The terms and conditions of the refund policy.

Appendix D ACH Transaction Confirmation (Online Check) Receipt

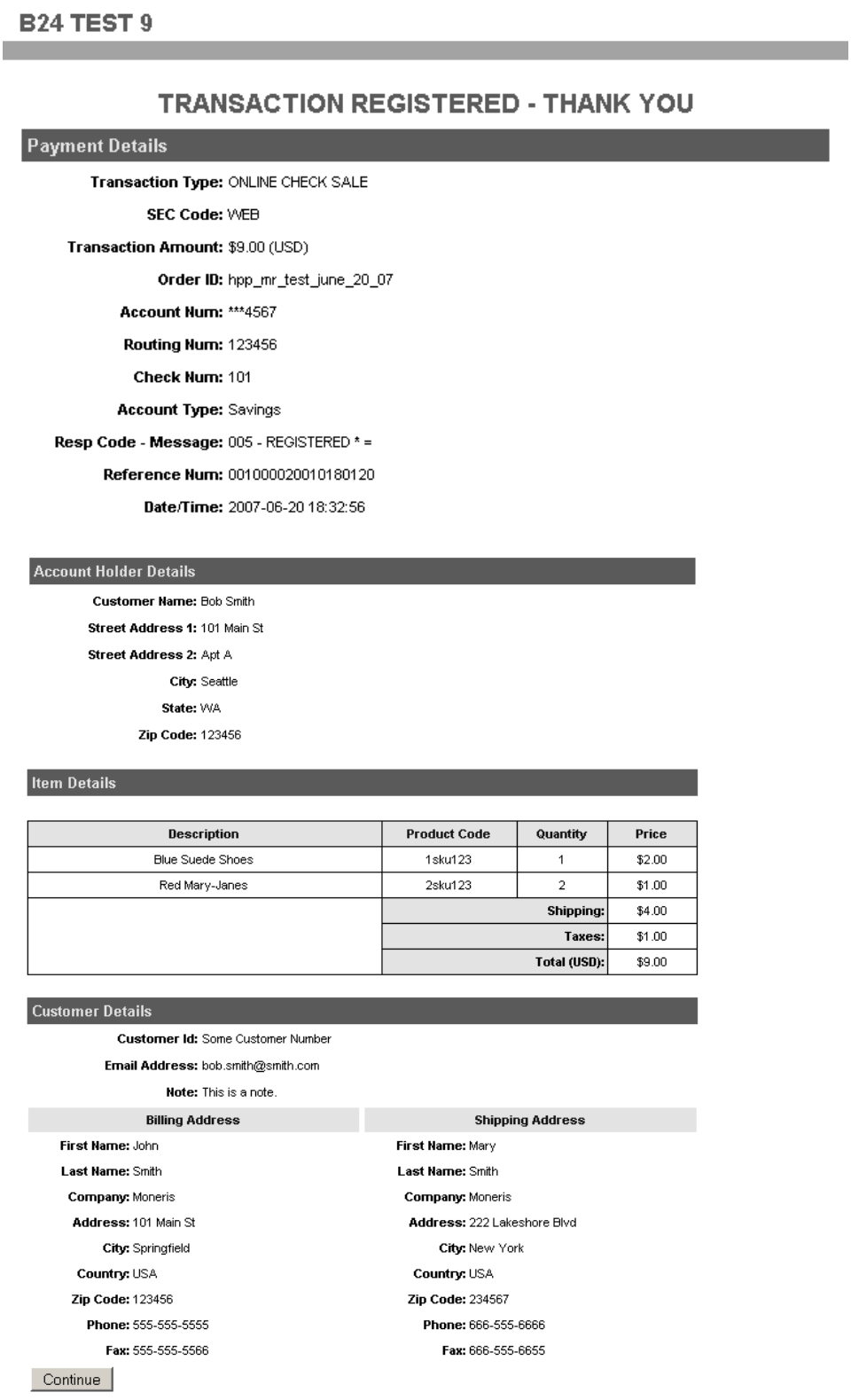
For an ACH transaction, a transaction confirmation is not mandatory; though Moneris Gateway does recommend that a receipt of registration of the transaction be provided to the customer. Below is a list of recommended fields, the format they are to be displayed in and the corresponding variable name as returned in a Hosted Payment Page GET or POST response.

	Field	Variable	Description
1	Merchant Name		The name of the store / business.
2	Merchant URL		Web site address of the store / business.
3	Transaction Type	txn_type	The type of transaction that was performed: • Online Check Sale (ACH Debit) **NOTE:** The term listed above is the name for the transaction as it is to be displayed on receipts. Other terms used for the transaction type are indicated in brackets.

	Field	Variable	Description
4	Payment Type	card_type	Indicates that this is an ACH processed transaction.
5	SEC Code		Specifies which SEC Code was submitted. Identifies how the bank account information was collected. Via the Hosted Paypage, the SEC code will be: WEB – Internet Initiated Entry
6	Transaction Amount	amount	The total amount being debited from the bank account.
7	Convenience Fee Amount	convenience_fee	The amount of the convenience fee. Please note, the 'Convenience Fee Amount' must be less than the 'Transaction Amount'.
8	Transaction Date and Time	txn_date txn_time	The date may be in any format, but must include the day, month and year. The time must be in 24 hour format. It is good practice to include the seconds in the time format to help with tracing transactions.

	Field	Variable	Description
9	Reference Number	ref_num	6400135 = terminal number 001 = shift 001 = batch number 001 = sequence number 0 = reserved
10	Auth Code	auth_code	The authorization number is only printed if the transaction is approved. If the transaction is declined the field may be omitted.
11	Response Code	response_code	The 3 digit response code returned in the transaction (ex. 000 – 999)
12	Response Message / Result	message	Message indicating whether the transaction was Registered or Failed to Register.
13	Account Number	card_num	Customer's bank account number. All but the last 4 digits of the account number must be masked out.

The sample below displays the layout of the response ACH / Online Check receipt. This transaction confirmation 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 color and style of the receipt, are determined by the settings chosen in the Hosted Payment Page configuration. Please refer to Hosted Payment Page Configuration Tool (see page 12) for more information.



B24 TEST 9

TRANSACTION REGISTERED - THANK YOU

Payment Details

Transaction Type: ONLINE CHECK SALE
SEC Code: WEB
Transaction Amount: $9.00 (USD)
Order ID: hpp_mr_test_june_20_07
Account Num: ***4567
Routing Num: 123456
Check Num: 101
Account Type: Savings
Resp Code - Message: 005 - REGISTERED * =
Reference Num: 001000020010180120
Date/Time: 2007-06-20 18:32:56

Account Holder Details

Customer Name: Bob Smith
Street Address 1: 101 Main St
Street Address 2: Apt A
City: Seattle
State: WA
Zip Code: 123456

Item Details

Description	Product Code	Quantity	Price
Blue Suede Shoes	1sku123	1	$2.00
Red Mary-Janes	2sku123	2	$1.00
		Shipping:	$4.00
		Taxes:	$1.00
		Total (USD):	$9.00

Customer Details

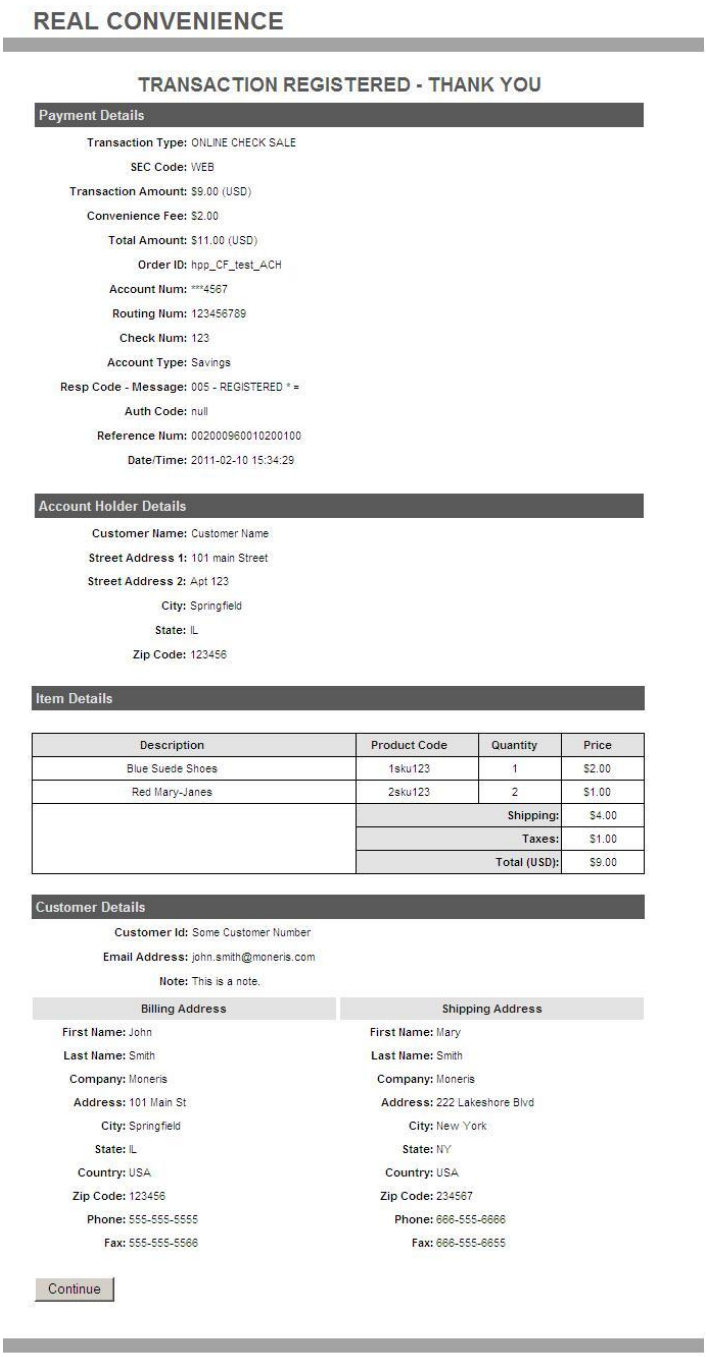
Customer Id: Some Customer Number
Email Address: bob.smith@smith.com
Note: This is a note.

Billing Address	Shipping Address
First Name: John	**First Name:** Mary
Last Name: Smith	**Last Name:** Smith
Company: Moneris	**Company:** Moneris
Address: 101 Main St	**Address:** 222 Lakeshore Blvd
City: Springfield	**City:** New York
Country: USA	**Country:** USA
Zip Code: 123456	**Zip Code:** 234567
Phone: 555-555-5555	**Phone:** 666-555-6666
Fax: 555-555-5566	**Fax:** 666-555-6655

Continue

The sample below displays the layout of the response ACH/Online Check receipt with Convenience Fee . This transaction confirmation 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 color and style of the receipt, are determined by the settings chosen in the Hosted Payment Pageconfiguration. Please refer to Hosted Payment Page Configuration Tool (see page 12) for further information.



REAL CONVENIENCE

TRANSACTION REGISTERED - THANK YOU

Payment Details

Transaction Type: ONLINE CHECK SALE
SEC Code: WEB
Transaction Amount: $9.00 (USD)
Convenience Fee: $2.00
Total Amount: $11.00 (USD)
Order ID: hpp_CF_test_ACH
Account Num: ***4567
Routing Num: 123456789
Check Num: 123
Account Type: Savings
Resp Code - Message: 005 - REGISTERED * =
Auth Code: null
Reference Num: 002000960010200100
Date/Time: 2011-02-10 15:34:29

Account Holder Details

Customer Name: Customer Name
Street Address 1: 101 main Street
Street Address 2: Apt 123
City: Springfield
State: IL
Zip Code: 123456

Item Details

Description	Product Code	Quantity	Price
Blue Suede Shoes	1sku123	1	$2.00
Red Mary-Janes	2sku123	2	$1.00
		Shipping:	$4.00
		Taxes:	$1.00
		Total (USD):	$9.00

Customer Details

Customer Id: Some Customer Number
Email Address: john.smith@moneris.com
Note: This is a note.

Billing Address	Shipping Address
First Name: John	**First Name:** Mary
Last Name: Smith	**Last Name:** Smith
Company: Moneris	**Company:** Moneris
Address: 101 Main St	**Address:** 222 Lakeshore Blvd
City: Springfield	**City:** New York
State: IL	**State:** NY
Country: USA	**Country:** USA
Zip Code: 123456	**Zip Code:** 234567
Phone: 555-555-5555	**Phone:** 666-555-6666
Fax: 555-555-5566	**Fax:** 666-555-6655

Continue

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 Gift Card Transaction Receipt

For all regular gift card transactions, the following fields must be presented to the cardholder on an approved or declined receipt. These required fields are listed below, including the corresponding variable names as returned in a Hosted Paypage POST containing XML response.

	Field	Variable	Description
1	Location description		The name of the store / business – this should be the same as what you have registered with Moneris Solutions.
2	Merchant URL		Web site address of the store / business.
3	Terminal ID	terminal_id	This field is also referred to as the ECR number (for example, 00016927). It is a 6-8 digit number that identifies the specific terminal that processed the transaction.
4	Transaction description	txn_type	The type of transaction that was performed: • Gift Purchase
5	Card description	card_desc	This field describes the gift card program.
6	Card number	card_num	The customer's copy

	Field	Variable	Description
			of the receipt must have all but the last 4 digits of card number masked out.
7	Transaction amount	total_amount	The total amount of the goods or services. This amount may be covered by only a gift card, multiple gift cards, or a combination of gift cards, loyalty card and a credit card.
8	Balance removed from the card	gift_charge_total	The total amount being debited or credited from to the gift card.
9	Remaining card balance	rem_balance	Optional. This amount is to be included if it is returned in the 'rem_ balance' field in the transaction response.
10	Reference number	ref_num	Optional. This field is to be included if it is returned in the 'ref_ num' field of the transaction response.
11	Receipt message	receipt_text	Optional. To be included if it is

	Field	Variable	Description
			returned in the 'receipt_text' field in the transaction response. Please note, the 'receipt_ text' field may include formatting codes that will outline how the text is to be printed on the receipt.
12	Date and time	date_time	The date may be in any format, but must include the day, month and year. The time must be in 24 hour format. It is good practice to include the seconds in the time format to help with tracing transactions.
13	Voucher text	voucher_text	Optional. To be included if it is returned in the 'voucher_text' field in the transaction response. Please note, the 'voucher_ text' field may include formatting codes that will outline how the text is to be printed on the receipt.
14	Goods and Services Order		A list of all items/services that are being paid for in this transaction.

	Field	Variable	Description
15	Return Policy		The terms and conditions of the refund policy.

Appendix G Loyalty Card Transaction Receipt

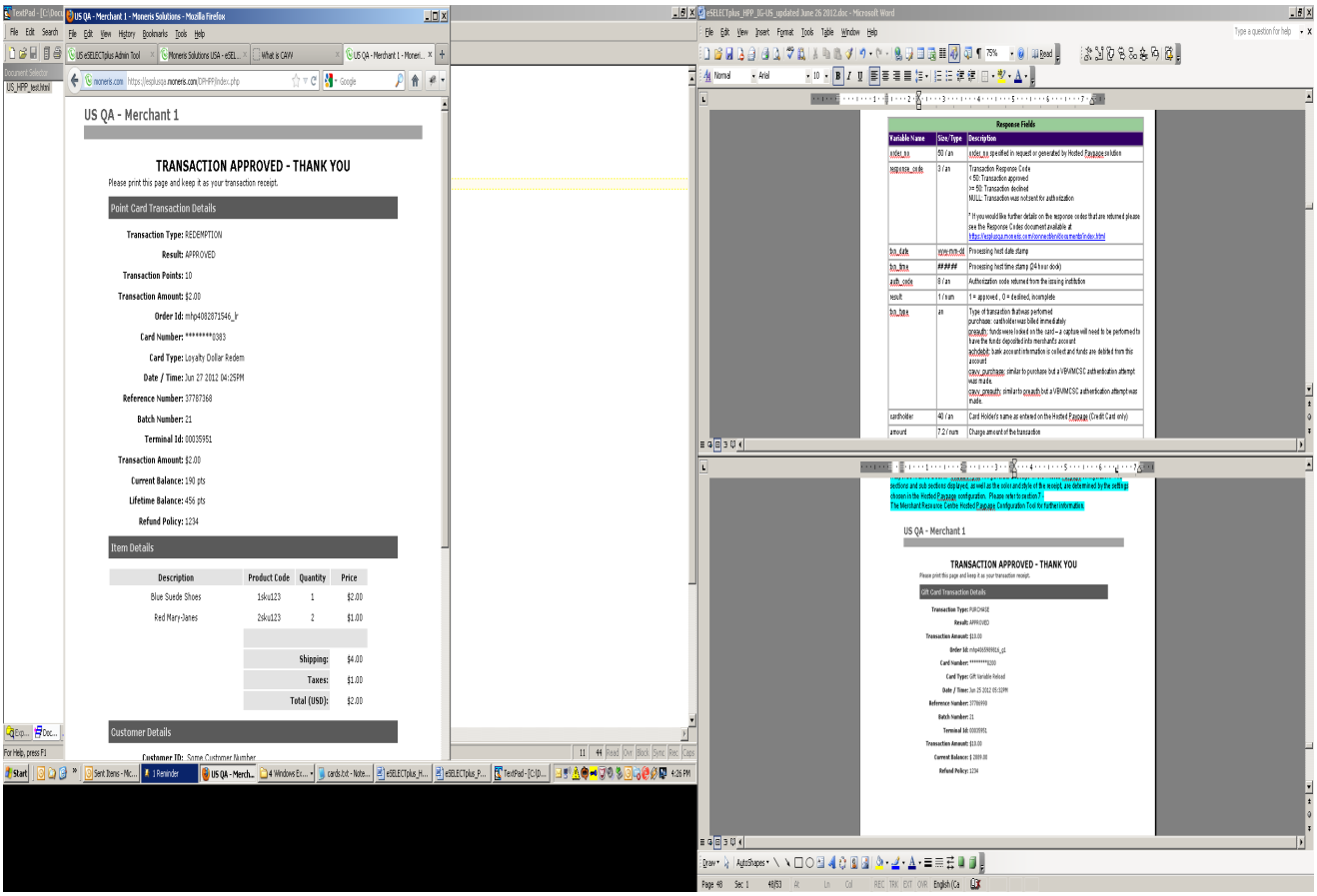
For all regular gift card transactions, the following fields must be presented to the cardholder on an approved or declined receipt. These required fields are listed below, including the corresponding variable names as returned in a Hosted Payment Page POST containing XML response.

	Field	Variable	Description
1	Location description		The name of the store / business; should be the same as what you have registered with Moneris Solutions.
2	Merchant URL		Web site address of the store / business.
3	Terminal ID	terminal_id	This field is also referred to as the ECR number (for example, 00016927). It is a 6-8 digit number that identifies the specific terminal that processed the transaction.
4	Transaction description	txn_type	The type of transaction that was performed: • Purchase • Redemption
5	Card description	card_desc	This field describes the gift card program.
6	Card number	card_num	The customer's copy of the receipt must have all but the last 4 digits of card number masked out.
7	Transaction amount	total_amount	The total amount of the goods or services. This amount may be covered by only a

	Field	Variable	Description
			loyalty card, or a combination of gift cards, loyalty card and a credit card.
8	Balance removed from the card	transaction_amount	Amount processed on this loyalty card transaction. This value will be displayed in dollars.
9	Current card balance	current_balance	Optional. This amount is to be included if it is returned in the 'current_balance' field in the transaction response.
10	Reference number	ref_num	Optional. This field is to be included if it is returned in the 'ref_ num' field of the transaction response.
11	Receipt message	receipt_text	Optional. To be included if it is returned in the 'receipt_text' field in the transaction response. Please note, the 'receipt_text' field may include formatting codes that will outline how the text is to be printed on the receipt.
12	Date and time	date_time	The date may be in any format, but must include the day, month and year. The time must be in 24 hour format. It is good practice to include the seconds in the time format to help with tracing transactions.

	Field	Variable	Description
13	Voucher text	voucher_text	Optional. To be included if it is returned in the 'voucher_text' field in the transaction response. Please note, the 'voucher_text' field may include formatting codes that will outline how the text is to be printed on the receipt.
14	Transaction Points	transaction_points	The total amount of points that were awarded/redeemed during the transaction
15	Lifetime Balance	lifetime_balance	This is the current lifetime balance for the card. If the value is non-zero, it should be printed on the receipt. Not all programs maintain a lifetime balance. Some programs use this field as an annual balance.
17	Return Policy		The terms and conditions of the refund policy.

The sample below displays the layout of the response credit card 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 sub-sections displayed, as well as the color and style of the receipt, are determined by the settings chosen in the Hosted Paypage configuration. Please refer to Hosted Payment Page Configuration Tool (see page 12) for further information.


US QA - Merchant 1
TRANSACTION APPROVED - THANK YOU
Please print this page and keep it as your transaction receipt.
Point Card Transaction Details
Transaction Type: REDEMPTION
Result: APPROVED
Transaction Points: 10
Transaction Amount: $2.00
Order Id: mhp4082871546_lr
Card Number: *********8383
Card Type: Loyalty Dollar Redem
Date / Time: Jun 27 2012 04:25PM
Reference Number: 37787368
Batch Number: 21
Terminal Id: 00035951
Transaction Amount: $2.00
Current Balance: 190 pts
Lifetime Balance: 456 pts
Refund Policy: 1234
Item Details
Description
Product Code
Quantity
Price
Blue Suede Shoes
1skus123
1
$2.00
Red Mary-Janes
2skus123
2
$1.00
Shipping:
$4.00
Taxes:
$1.00
Total (USD):
$2.00
Customer Details
Response Fields
Variable Name
Size/Type
Description
order_no
50 / an
order_no specified in request or generated by Hosted Paypage in Action
response_code
3 / an
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 at
https://esplus.moneris.com/manuals/docs/merchDirect/index.html
txn_date
yyyy-mm-dd
Processing host date stamp
txn_time
####
Processing host time stamp (24 hour clock)
auth_code
8 / an
Authorization code returned from the issuing institution
result
1 / num
1 = approved , 0 = declined, incomplete
txn_type
an
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
ach_debit: bank account information is collect and funds are debited from this account
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.
cardholder
40 / an
Card Holder's name as entered on the Hosted Paypage (Credit Card only)
amount
7.2 / num
Change amount of the transaction

Appendix H Gift Card/Loyalty Card Text Formatting Codes

There are a number of fields returned in a Gift Card or Loyalty Card receipt when selecting to have the response sent back to your server in a POST containing XML. These formatting codes pertain specifically to the Receipt Text and Voucher Text fields outlining how to display the text properly on the cardholder's receipt.

Each text field has a maximum length. Within the maximum length, the text message can be organized into many lines (up to a maximum of 10). Each line is terminated with a line feed character (0x0A). The last line of the message may not include a line feed.

The first character or each line may be a formatting code. The merchant should then act on the formatting code and print all text up to the next line feed (or end of message) according to the specified formatting code. The following formatting codes are defined:

0x07 - Normal height, left justified

0x08 - Normal height, centred

0x0F - Normal height, right justified

0x0B - Double height, left justified

0x0C - Double height, centred

0x0E - Double height, right justified

If a formatting code is not included in the message text, or if the merchant is unable to print using the specified format, then the merchant may choose their own default format.

Appendix I 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>
```