Developer Guide To The

Virtual Merchant

March 16, 2007

THIS VIRTUAL MERCHANT DEVELOPER'S GUIDE
WILL FAMILIARIZE YOU WITH ALL THE TRANSACTION
TYPES AND PROCEDURES YOU WILL USE. PLEASE TAKE SOME
TIME TO THOROUGHLY REVIEW THIS MATERIAL

NOVA NETWORK

The most reliable payment processing
network in the industry

Table of Contents

General Overview

Virtual Merchant is a payment gateway allowing merchants to submit transactions via the Internet to the NOVA Network™ and have them authorized in real-time. This guide is designed to help you, the developer, write a Website that will interface with Virtual Merchant to process these transactions. Virtual Merchant allows for a wide range of customization by utilizing both code and configuration through the administration section of your account at www.myvirtualmerchant.com, to allow you to process transactions in the manner that best fits your particular business needs.

Virtual Merchant accepts information sent via HTTPS, either by a Get (along with the Query string in the URL) or Post (typically from an HTML Form) Method. The information you send, along with your Virtual Merchant settings, will determine how your transactions are handled. This information will determine user interface settings such as colors, images to display, field order, and field selection. This information also will determine behavior. For example, if Virtual Merchant needs to display a form to collect additional customer information, the information you send and your Virtual Merchant settings will determine the type of receipt to display to the customer, and how information is sent back to you.

The first decision you need to make is how much work you want to do, and how much work you want to let Virtual Merchant do for you. Virtual Merchant can accept as little as two pieces of data from your Website, and do the rest of the work on its own, by gathering information from your customer and using the settings that have been configured by your company in the Virtual Merchant administration section. On the other hand, you could go as far as using Virtual Merchant as a backend feature to your Website, completely transparent to your customers, in which you write the page that gathers all of the pertinent customer information and the receipt page that displays the outcome of the transaction processing to the user. We find that most merchants fall somewhere in the middle of these scenarios, gathering some data from their customer's before sending them out to Virtual Merchant and then letting Virtual Merchant gather more information from the customer and displaying the receipt after credit card approval.
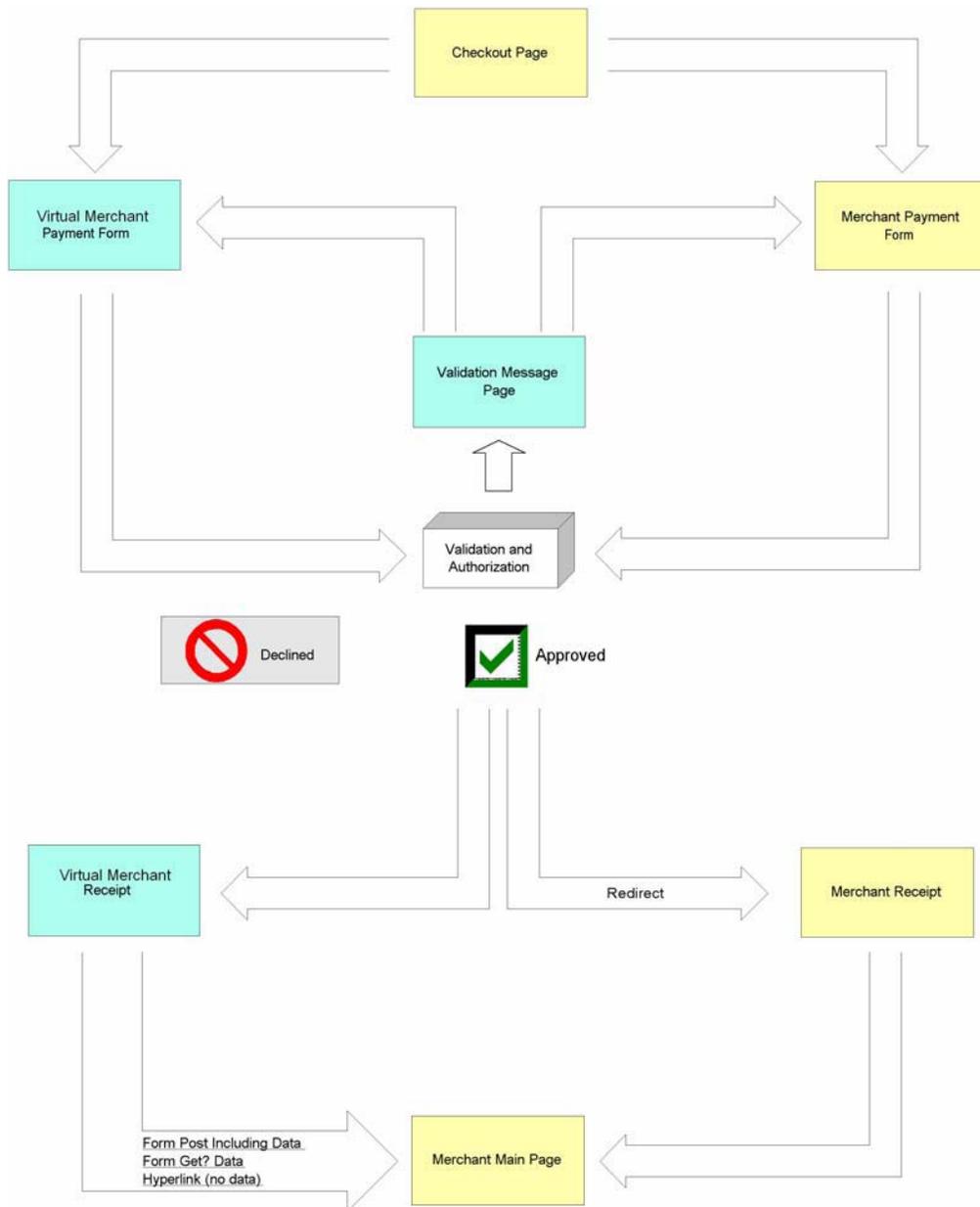
The Virtual Merchant program is easy to integrate with other payment systems. For more information on integration, please call **1-800-377-3962** or e-mail your question to: internetproductsupport@merchantconnect.com.

Flowchart

In this Developer's Guide, flowchart diagrams are used for visual aides to help you maintain your bearings as you navigate through the transaction process.

The blue boxes represent the Virtual Merchant process in the chart and the yellow boxes represent the merchant side of the transaction.

The flowchart will appear in its entirety throughout the document. You'll be able to determine your location in the transaction, as the corresponding section of the flowchart will appear in color. All other quadrants of the chart will appear shaded. The green checkmark in the middle represents an approved transaction and the red, crossed circle represents a declined transaction.
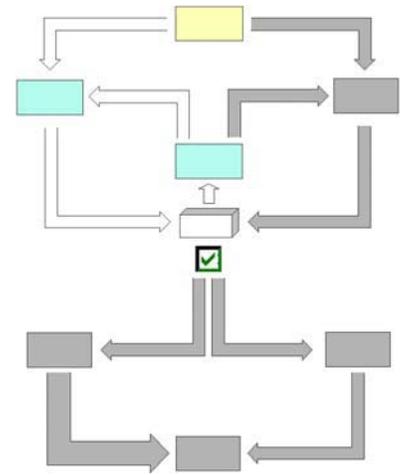
Payment Form

The Payment Form is the place where your customer enters the necessary and/or required personal and credit card information required to process the transaction. It also is the page that sends the transaction on to the Virtual Merchant system for authorization processing. If Virtual Merchant is drawing the Payment Form for you, you will only need to give the system enough information to know who you are and any special information about your transaction that the customer is not going to be entering. If you are drawing your own payment form, then your form will need to send all of the necessary data to complete the transaction into Virtual Merchant.

Virtual Merchant Provided Form

This section will tell you what you need to do to send information to Virtual Merchant in order to have Virtual Merchant present a payment form to your customer. This payment form will gather information from your customer such as the name displayed on their credit card, card number, expiration date, billing and shipping address, as well as other fields specified by you in your Web page's code or in the Terminal Setup section of your Virtual Merchant account.

Input

The first, and most basic, step is to submit the minimum information to Virtual Merchant. The minimum information required to provide a payment form to your customer is the following two fields:

- Your ssl_merchant_id (sometimes referred to as your Account ID or your Virtual Merchant ID)
- ssl_PIN that matches your Account to the terminal you are using.

If you have more than one terminal assigned to your account, you need to make sure that the PIN you are using corresponds to the correct terminal. With these two pieces of information, Virtual Merchant can display a payment form based on the settings you have pre-determined in your Virtual Merchant account that will allow your customers to enter all of the transaction data including the amount. Although in most cases this is not practical, this could be used to accept online donations, for which the customer decides the amount of money he/she wishes to spend.

In a more likely scenario you will have some additional information to send to Virtual Merchant, including the transaction amount, a description, and perhaps other fields such as a customer code, or an invoice number. Virtual Merchant would then draw a payment form for your customer to enter their credit card information, and other information such as their billing address and any custom fields that you have created.

To integrate Virtual Merchant with a Website that offers paid goods or services and wants to charge for those goods or services by credit card, the following procedure can be followed:

1. Create a form on your Website
2. Set the action of the form to https://www.myvirtualmerchant.com/VirtualMerchant/process.do
3. Set the method of the form to POST
4. Add a hidden field with the name ssl_merchant_ID. Set the value to the Virtual Merchant account ID.
5. Add a hidden field with the name ssl_pin. Set the value to the merchant PIN associated with the Virtual Merchant ID.
6. Add a hidden field with the name ssl_amount. Set the value to the desired amount.
7. Add a submit button.

Example 1:

The following HTML code demonstrates the initiation of a minimal transaction, in which Virtual Merchant gathers all of the customer's billing information:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_amount" value="14.95">
 <input type="submit" value="Click To Order">
</form>
```

This code creates a button with the label "Click To Order". When the button is clicked, the customer is taken to the payment form on the secure Virtual Merchant servers.

Please Note: In all of these examples, you will have to change the data values, such as "my_virtualmerchant_ID", "my_PIN", and the amount of "14.95" to values that match your Virtual Merchant account and meet your Website's needs.

Each account will have one Merchant Admin user and can have multiple Employee Users, who may or may not have access to run transactions on the terminal that is integrated with your Website. When specifying a User ID, make sure that the submitted PIN matches the User ID that you are submitting and the terminal on which you wish to run the transaction. When an account has more then one Terminal, it is the combination of ssl_merchant_id, ssl_pin and ssl_user_id that Virtual Merchant will use to determine which of the terminals the transaction is being submitted to. The terminal itself will not be submitted to Virtual Merchant. When ssl_user_id is omitted, the User ID is assumed to be the same as the ssl_merchant_id, the merchant Admin User.

Example 2:

The following HTML code demonstrates the initiation of a slightly more complex transaction in which the Virtual Merchant system will display a form to gather additional required customer information:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_show_form" value="true"> <!-- Optional, Default Value = "true" -->
 <input type="hidden" name="ssl_test_mode" value="false"> <!-- Optional, Default Value = "false" -->
 <input type="hidden" name="ssl_invoice_number" value="Inv-123-ABC">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_amount" value="31.01">
 <input type="hidden" name="ssl_salestax" value="1.86">
 <input type="submit" value="Click Here to Complete Your Order">
</form>
```

This code creates a button with the label "Click Here to Complete Your Order." When the button is clicked, the customer is taken to the payment form on the secure Virtual Merchant servers to fill in the final details of their transaction.

Behavior

In both of these examples, the output is an HTML page from Virtual Merchant displayed on the customer's Web Browser. This page will either be a Payment Form to enter more information or an error message, indicating an error condition with the data sent to Virtual Merchant (such as when the ssl_PIN doesn't match an existing terminal for the ssl_merchant_ID). See Appendix 3 for more about the Error Messages and Error Numbers.

Your customer will use the Payment Form that is displayed to provide the final pieces of information needed to complete the transaction. This final information will include at least the credit card number and expiration date. Most likely, other information such as Billing Address, Shipping Address, and e-mail will be included as well. The fields that are presented on the Virtual Merchant payment form are fully configurable from within the Virtual Merchant site itself. Please see the User's Guide for more information on how to do this.
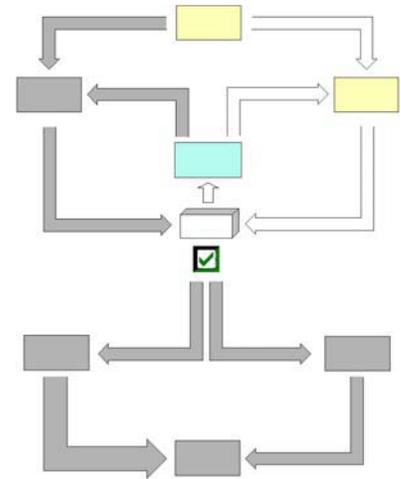
Merchant Provided Form

This section explains how to send information in order to have Virtual Merchant process a credit card transaction without additional input from your customer. Using this method, Virtual Merchant will only be visible to your customer before the transaction is processed if the data passed in is found to be invalid.

Invalid data could consist of data such as a bad PIN or User ID that does not allow the transaction to be associated with your account, or it could bad customer data such as a mal-formed credit card number or e-mail address.

Input

If you want to collect all of the data from the customer on your Website, and only send the information to Virtual Merchant after it has all been gathered you can do so. To hide the Payment Form you will need to send the parameter "ssl_show_form" with a value of "False".

When presenting the Payment form and not asking your customers to input any additional fields on a Virtual Merchant provided Web page, you must include some additional information. The three additional required fields that you must pass to Virtual Merchant are:

- ssl_amount
- ssl_card_number
- ssl_exp_date

Example 3:

This example shows a very basic form that collects and passes the minimum required data for a complete Virtual Merchant transaction that will not display the Virtual Merchant payment form:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00">
 <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_show_form" value="false">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4"> <br/>
 <br/>
 <input type="submit" value="Continue">
</form>
```

This code creates a form displaying the customer's total, and asking for their credit card number and expiration date, with a button labeled "Continue." After the user enters the information and the button is clicked, the transaction is processed by the secure Virtual Merchant servers. Then the user is taken directly to a receipt or result form, displaying the outcome of the transaction.

Please Note: In all of these examples, you will have to change the data values, such as "my_virtualmerchant_ID", "my_PIN", and the amount of "5.00" to values that match your Virtual Merchant account and meet your Website's needs.

Each account can have one Merchant Admin user and can have multiple Employee Users, who may or may not have access to run transactions on the terminal that is integrated with your Website. When specifying a User ID, please make sure that the PIN that is submitted matches up with the User ID that you are submitting and the terminal on which you wish to run the transaction. When an account has more then one Terminal, it is the combination of ssl_merchant_id, ssl_pin, and ssl_user_id that Virtual Merchant will use to determine which of the terminals the transaction is being submitted to. The terminal itself will not be submitted to Virtual Merchant. When ssl_user_id is omitted, the User ID is assumed to be the same as the ssl_merchant_id, the merchant Admin User.

Using Address Verification Service (AVS)

AVS is a service that is performed by the credit card issuer. It compares the street address and zip code supplied with the credit card number to the address and zip code on file. The following three system fields are used by Virtual Merchant to handle the AVS:
- ssl_avs_address
- ssl_avs_zip
- ssl_avs_response

The first two fields are visible by default on the payment form under the names "Address" and "Zip Code" and control the input to the AVS. If you are collecting address information on your Website, or already have this information in a customer database, before displaying the Virtual Merchant form, you can pass the information to Virtual Merchant as variables in the same fashion that the PIN and ID's are sent. The last field contains the AVS response and is stored with the transaction, as well as included in the receipt.

Virtual Merchant may also return a space, in which case AVS was not performed. The most likely reasons for this are:
- No address or zip code was supplied
- AVS not supported by card issuer

See Appendix 2: AVS Response Codes for more detail.
Please Note: A failed AVS check will NOT result in a Decline response for the transaction itself.

Using CVV2/CVC2

CVV2 and CVC2 are fraud-prevention measures implemented by VISA and MasterCard. These numbers are located on the back of the credit card and are generally three- or four-digit numbers listed after the credit card number. CVV2 refers to the VISA number and CVC2 refers to MasterCard's implementation.

By providing this number, virtual (Internet) cardholders can help prove they have the physical card in hand when completing the transaction. This helps eliminate the possibility of someone using a stolen card number on your Website.
The following three system fields are used by Virtual Merchant to handle CVV2/CVC2:
- ssl_cvv2
- ssl_cvv2cvc2
- ssl_cvv2_response

VISA implements two fields when using CVV2: the CVV2 Indicator (ssl_cvv2) and the CVV2 Data (ssl_cvv2cvc2). The indicator is one of the following:
- Present
- Bypassed
- Not Present
- Illegible

The CVV2 Data indicates the actual number on the card. When passing the CVV2 data, you must also pass the CVV2 indicator, or the transaction will be rejected by Virtual Merchant.

MasterCard does not implement an indicator like VISA. The CVC2 data (ssl_cvv2cvc2) is all that is necessary. When you pass a MasterCard number, CVV2 indicator and CVC2 data, the indicator is ignored. This is by design.

See Appendix 2: CVV2/CVC2 Response Codes for more detail.
Please Note: A failed CVV2/CVC2 check should result in a Decline response for the transaction.

Example 4:

To pass AVS data and CVV2 data or CVC2 data, use the following syntax:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_card_number" value="0000000000000000">
 <input type="hidden" name="ssl_exp_date" value="0000">
 <input type="hidden" name="ssl_amount" value="12.77">
 <input type="hidden" name="ssl_show_form" value="false">
 <input type="hidden" name="ssl_cvv2" value="present">  <!--CVV2 Indicator -->
 <input type="hidden" name="ssl_cvv2cvc2" value="1234">  <!--CVV2 Data -->
 <input type="hidden" name="ssl_avs_address" value="123 Main St.">  <!--AVS Postal Address -->
 <input type="hidden" name="ssl_avs_zip" value="01234">  <!--AVS ZIP Code -->
 <input type="submit" value="Donate Now">
</form>
```

Behavior

In both Example 3 and Example 4, the output will be an HTML Receipt page describing the outcome of the transaction displayed to the customer. This page will either be a receipt or result page drawn by Virtual Merchant, an error message indicating an error condition with the data sent to Virtual Merchant

(such as when the ssl_PIN doesn't match an existing terminal for the ssl_merchant_ID), or a Receipt page developed by you. See Appendix 3 for more about the Error Messages and Error Numbers and see the next two chapters for more information on Receipt pages.

Receipt Options

A receipt is the customer's documentation of the outcome of the transaction authorization. It also is the page that can send the transaction information back to your Website, based on the customer's actions. If Virtual Merchant is drawing the receipt for you, then you not need to include logic to parse through the Virtual Merchant result, but your customer might not return to your Website when the transaction is complete. If you are drawing your own receipt, then your form will need to handle the data received from Virtual Merchant in order to correctly communicate to your customer the outcome of their transaction.

Virtual Merchant Provided Receipt

This section instructs you on the methodology of allowing Virtual Merchant to display the receipt to your customer. The receipt has quite a few configuration possibilities that can be driven by code or by choices made in the Administration section of the Virtual Merchant Website.
Please see the User's Guide for more information on how to use the
Virtual Merchant Website to configure your receipt options.

Input

There are four primary variables that dictate how receipts are processed:

- ssl_result_format
- ssl_receipt_link_method
- ssl_receipt_link_url
- ssl_receipt_link_text

You also have the option to split the last three of these up to allow for a different type of receipt for Approvals vs. Declines. If you use the variables above, they will take precedence over the following parameters:

- ssl_receipt_decl_method
- ssl_receipt_decl_get_url
- ssl_receipt_decl_post_url
- ssl_receipt_decl_text
- ssl_receipt_apprvl_method
- ssl_receipt_apprvl_get_url
- ssl_receipt_apprvl_post_url
- ssl_receipt_apprvl_link_text

ssl_result_format has acceptable values: ASCII and HTML. If you do not specify the format, an HTML receipt will be returned. If you specify ASCII, only a list of key/value pairs will be returned, the other receipt related parameters you have sent in would be ignored. The ASCII format is intended to be called on by a separate application that will process the data, instead of directly by a Web page used by customers initiating transactions.
There are four options for the various ssl_receipt_link_method variables:

- GET
- POST
- LINK
- REDG (REDirect GET)

The first two choices utilize the button at the bottom of the receipt for the customer to select so that they may return to your Website. These two options will pass the transaction's data back to your site via the

method chosen. LINK presents a hyperlink at the bottom of the Virtual Merchant receipt page, and does not transmit data back to your Website. REDG (RE-Direct Get) is covered in more detail in the next section.

Example 5:

To pass receipt options, use the following syntax:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
  Your Total: $5.00 <br/>
  <input type="hidden" name="ssl_amount" value="5.00">
  <br/>
  <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
  <input type="hidden" name="ssl_user_id" value="my_User_ID">
  <input type="hidden" name="ssl_pin" value="my_PIN">
  <input type="hidden" name="ssl_show_form" value="false">
  <input type="hidden" name="ssl_transaction_type" value="ccsale">
  <input type="hidden" name="ssl_invoice_number" value="123-ABC">
  <input type="hidden" name="ssl_email" value="test@test.com">
  Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
  Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4"> <br/>
  <br/>
  <input type="hidden" name="ssl_result_format" value="HTML">
  <input type="hidden" name="ssl_receipt_decl_method" value="POST">
  <input type="hidden" name="ssl_receipt_decl_post_url" value="http://www.website.com/decline.asp">
  <input type="hidden" name="ssl_receipt_apprvl_method" value="GET">
  <input type="hidden" name="ssl_receipt_apprvl_get_url" value="http://www.website.com/approval.asp">
  <input type="hidden" name="ssl_receipt_link_text" value="Continue">
  <input type="submit" value="Continue">
</form>
```
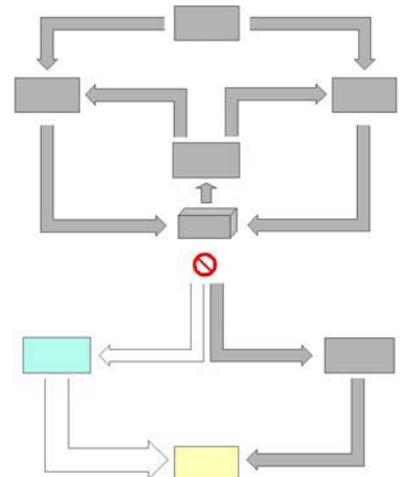
This generates a receipt that includes the following code for an accepted transaction:

```
This is your Receipt<br><br>
…
<!--The visible portion of your receipt will appear here, according to the configuration settings you applied
in the Virtual Merchant administrative Website.-->
…
<form action="http://www.website.com/approval.asp" method="GET">
<input type="hidden" name="ssl_result" value="0">
<input type="hidden" name="ssl_result_message" value="APPROVED">
<input type="hidden" name="ssl_txn_id" value="99C7884A-EDB6-4256-BE69-4547B8859D5B">
<input type="hidden" name="ssl_approval_code" value="N29032">
<input type="hidden" name="ssl_cvv2_response" value="">
<input type="hidden" name="ssl_avs_response" value=" ">
<input type="hidden" name="ssl_transaction_type" value="ccsale">
<input type="hidden" name="ssl_invoice_number" value="123-
ABC">
<input type="hidden" name="ssl_amount" value="5.00">
<input type="hidden" name="ssl_email" value=" test@test.com">
<br>
<input type="submit" value="Continue" class="smallbutton">
</form>
```

Or the result is a form that includes the following code for a declined transaction:

```
<b>An Error Occurred</b><br><br>
      Number : 1<br>
      Message : This transaction request has not been approved.  You may elect to use another form of
payment to complete this transaction or contact customer service for additional options.<br>
<form action="http://www.website.com/decline.asp" method="POST">
<input type="hidden" name="ssl_result" value="1">
<input type="hidden" name="ssl_result_message" value="DECLINED">
<input type="hidden" name="ssl_txn_id" value="B6637C93-CA38-41C5-951A-C995BFFBD708">
<input type="hidden" name="ssl_approval_code" value="      ">
<input type="hidden" name="ssl_cvv2_response" value="">
<input type="hidden" name="ssl_avs_response" value=" ">
<input type="hidden" name="ssl_transaction_type" value="ccsale">
<input type="hidden" name="ssl_invoice_number" value="123-ABC">
<input type="hidden" name="ssl_amount" value="5.00">
<input type="hidden" name="ssl_email" value=" test@test.com">
<br>
<input type="submit" value="Continue" class="smallbutton">
</form>
```

Please Note: In all of these examples, you will have to change the data values, such as "my_virtualmerchant_ID", "my_PIN", "123-ABC"and the amount of "5.00" to values that match your Virtual Merchant account and meet your Website's needs.

Behavior

An HTML page will display informing whether the transaction was approved or not. If the transaction was approved, then the receipt displays the data elements that make up the transaction. A link back to your Website is displayed at the bottom of the page. This link is configured based on the parameters you set or by the configuration settings applied in the Virtual Merchant administrative Website. If you choose, you can set the format to ASCII or override the receipt link parameter in your code. Also, it is possible to specify the behavior for the approvals separate from the behavior of the declines.

Example 5, above, shows two possible outcomes for a Virtual Merchant Receipt. **A receipt containing ssl_result=0 represents an Approved transaction. A receipt containing any other value for ssl_result represents a Declined transaction, or a transaction that had an error, preventing it from being authorized.** See Appendix 2 for other possible response codes.
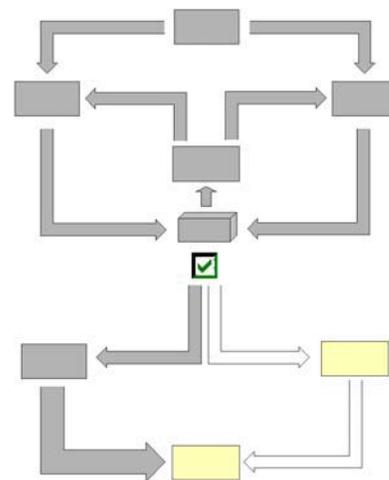
Merchant Provided Receipt

This section explains what you need to do to show your customer a receipt of your own creation for a Virtual Merchant transaction. The receipt has quite a few configuration possibilities that can be driven by code, or by choices made in the Administration section of the Virtual Merchant Website. Please see the User's Guide for more information on using the Virtual Merchant Website to configure your Receipt options.

Input

There are four primary variables that dictate how Receipts are processed:

- ssl_result_format
- ssl_receipt_link_method
- ssl_receipt_link_url

- ssl_receipt_link_text

You also have the option to split the last three of these up to allow for a different type of receipt for Approvals vs. Declines. If you use the variables above, they will take precedence over the following parameters:

- ssl_receipt_decl_method
- ssl_receipt_decl_get_url
- ssl_receipt_decl_post_url
- ssl_receipt_decl_text
- ssl_receipt_apprvl_method
- ssl_receipt_apprvl_get_url
- ssl_receipt_apprvl_post_url
- ssl_receipt_apprvl_link_text

Result Format

Ssl_result_format has acceptable values:

- ASCII
- HTML

If you do not specify the result format, an HTML receipt will be returned. If you select ASCII, only a list of key/value pairs will be returned, the other Receipt related parameters you have set are ignored. The ASCII format is recommended if you are using an intermediary application to send transactions to Virtual Merchant rather than sending transactions directly from an HTML form on a Web page that is driven by your customer's actions. The ASCII format will allow you to easily parse through the transaction data and choose what to display to your customer, and what data to use in other ways for your own application.

Example 6:

To pass receipt options requesting an ASCII response, use the following syntax:

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00">
 <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_show_form" value="false">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_invoice_number" value="123-ABC">
 <input type="hidden" name="ssl_email" value="test@test.com">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4"> <br/>
 <br/>
 <input type="hidden" name="ssl_result_format" value="ASCII">
 <input type="submit" value="Continue">
</form>
```

Producing a receipt that includes the following key/value pairs for an Accepted transaction:

```
ssl_result=0
ssl_result_message=APPROVED
ssl_txn_id=9621F9AD-E49E-4003-91BD-5C1B08569959
ssl_approval_code=N54032
ssl_cvv2_response=
ssl_avs_response=
ssl_transaction_type=ccsale
ssl_invoice_number=123-ABC
ssl_amount=5.00
ssl_card_number=00*******0000
ssl_exp_date=0000
ssl_email=test@test.com
```

And producing a receipt that includes the following key/value pairs for a Declined transaction:

```
ssl_result=1
ssl_result_message=This transaction request has not been approved. You
may elect to use another form of payment to complete this transaction or
contact customer service for additional options.
```

Example 6 (above) shows 2 possible outcomes for an ASCII Virtual Merchant Receipt. **A receipt containing ssl_result=0 represents an Approved transaction. A receipt containing any other value for ssl_result represents a Declined transaction, or a transaction that had an error, preventing it from being authorized.**

Receipt Link Method: Re-direct Get

There are four options for the various ssl_receipt_link_method variables. To display a receipt of your own you must use REDG (RE-Direct Get). REDG will redirect the customer's Browser to the URL of your choosing as soon as the transaction is processed by Virtual Merchant.

Using the various ssl_receipt_link_url variables Virtual Merchant gives you the option of sending approved and declined transactions to the same URL or to different URLs to handle them separately. If you are using the REDG method and wish to have separate approved and declined behaviors, you will want to use the "get" versions of the ssl_receipt_link_url variables, to specify the destination URL. Specifically:
- ssl_receipt_decl_get_url
- ssl_receipt_apprvl_get_url

Example 7:

To pass receipt options for redirecting the customer to your own receipt, use the following syntax:
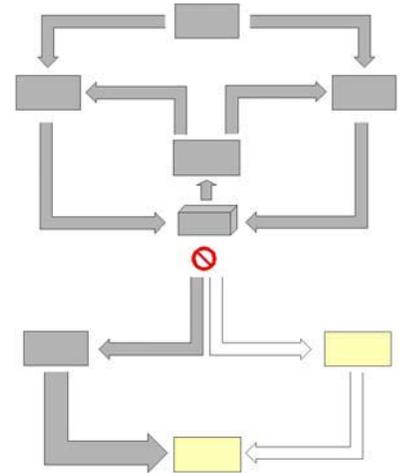
```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00"> <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_show_form" value="false">
 <input type="hidden" name="ssl_invoice_number" value="123-ABC">
 <input type="hidden" name="ssl_email" value="test@test.com">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4"> <br/>
 <br/>
 <input type="hidden" name="ssl_result_format" value="HTML">
 <input type="hidden" name="ssl_receipt_decl_method" value="REDG">
 <input type="hidden" name="ssl_receipt_decl_get_url" value="http://www.website.com/decline.asp">
 <input type="hidden" name="ssl_receipt_apprvl_method" value="REDG">
 <input type="hidden" name="ssl_receipt_apprvl_get_url" value="http://www.website.com/approval.asp">
 <input type="submit" value="Continue">
</form>
```

Redirecting to "http://www.website.com/approval.asp" for an Accepted transaction or to "http://www.website.com/decline.asp" for a Declined transaction. The transaction data will be passed along as Get variables in the query string of the URL.

Behavior

In examples 6 and 7, your customer will see the receipt page you provide. Either your application will have received the transaction data in an ASCII format, and must then re-present that information to your customers in whatever fashion you determine, or their browser will be redirected to the page you have specified with transaction data being passed in the query string.

_____

Example 8 – Electronic Gift Card

In this example you will see the differences in submitting an EGC transaction to our system. The code below is for a gift card sale transaction.   For Activation and Reload transactions, the ssl_egc_tender_type field will need to be submitted. (see appendix 1 for ssl_egc_tender_type values).

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00">
 <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="egcsale">
 <input type="hidden" name="ssl_invoice_number" value="123-ABC">
 <input type="hidden" name="ssl_email" value="test@test.com">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4">
 <input type="submit" value="Continue">
</form>
```

Example 9 – PinLess Debit

In this example, PinLess Debit transaction is run. This type of transaction is almost the same as examples 1-3 with a couple of fields needed in addition, ssl_customer_number and ssl_account_type. This will authorize the card as a debit transaction.

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00">
 <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="pldpurchase">
 <input type="hidden" name="ssl_invoice_number" value="123-ABC">
 <input type="hidden" name="ssl_email" value="test@test.com">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4">
 Customer Number: <input type="text" name="ssl_customer_number"> <br/>
 Account Type <select name = ssl_account_type>
        <option value = "1" >Checking</option>
        <option value = "0" >Savings</option>
        </select> <br/>
 <input type="submit" value="Continue">
</form>
```

Example 10 – Dynamic Currency Conversion

If you participate in the DCC program, the following example should be reviewed. If you set the field ssl_result_format to HTML, VirtualMerchant will handle transactions submitted to process.do. The initial submission to process.do will be the same as in examples 1-3. The transaction is not complete at this point. The customer will be taken to a DCC decision page where they will need to decide what currency they want to process the transaction in. Once this is selected the transaction will complete as normal. The DCC decision page will appear if the customer's card is identified as a foreign currency currently supported. The post would be submitted as normal.

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 Your Total: $5.00 <br/>
 <input type="hidden" name="ssl_amount" value="5.00">  <br/>
 <input type="hidden" name="ssl_merchant_id" value="my_virtualmerchant_ID">
 <input type="hidden" name="ssl_user_id" value="my_User_ID">
 <input type="hidden" name="ssl_pin" value="my_PIN">
 <input type="hidden" name="ssl_transaction_type" value="ccsale">
 <input type="hidden" name="ssl_show_form" value="false">
 <input type="hidden" name="ssl_invoice_number" value="123-ABC">
 <input type="hidden" name="ssl_email" value="test@test.com">
 Credit Card Number: <input type="text" name="ssl_card_number"> <br/>
 Expiration Date (MMYY): <input type="text" name="ssl_exp_date" size="4"> <br/>
 <br/>
 <input type="hidden" name="ssl_result_format" value="HTML">
 <input type="hidden" name="ssl_receipt_decl_method" value="REDG">
 <input type="hidden" name="ssl_receipt_decl_get_url" value="http://www.website.com/decline.asp">
 <input type="hidden" name="ssl_receipt_apprvl_method" value="REDG">
 <input type="hidden" name="ssl_receipt_apprvl_get_url" value="http://www.website.com/approval.asp">
 <input type="submit" value="Continue">
</form>
```

The DCC Decision page will appear after process is selected. The customer will need to select one of the buttons to continue.

SALE - Dynamic Currency Confirmation	
Transaction Currency	EUR
Conversion Rate	1.3028
Total (USD)	1.00
Total (EUR)	1.3028

Please charge my purchase in my home currency	Do not charge me in my home currency; charge my purchase in the foreign currency

All currency choices are final.

ASCII Integration

If you set the ssl_result_format field to ASCII it is the merchant's responsibility to provide the above page to the customer. There is a 10 minute window to return a response. If the request is sent after 10 minutes the user will receive an error in response indicating the transaction was timed out and will have to begin the transaction again. If the response is received within the 10-minute time frame the transaction will complete and provide the authorization response message as in examples 1-3. Any error including unknown/missing ID or timeout would be directed to the error url, if one is specified. If no error URL is specified, the error will follow the normal response pattern (approval/decline response) for that particular

Option. A separate URL for DCC decisions can be specified. The variable is called ssl_dcc_popup_url. There are 4 different ways to integrate this each listed as an option below

OPTION A (receipt option LINK, GET, or POST, no ssl_dcc_popup_url defined)
Example response data string

```
id=PBczyFq67o8Vfzx9LdPyPKS&ssl_txn_currency_code=CAD&ssl_conversion_rate=1.2345&ssl_amount
=1.00&ssl_cardholder_amount=1.23&dccoption={(option label='Please charge my purchase in my home
currency')=Y;(option label='Do not charge me in my home currency; charge my purchase in US
dollars')=N}
```

Data sent to process.do
Example

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 <input type="hidden" name="id" value=" PBczyFq67o8Vfzx9LdPyPKS ">
 <input type="hidden" name="dccoption" value="Y">
 <input type="submit" value="Continue">
</form>
```

process.do response will be as in earlier examples.

END OPTION A

process.do response:

OPTION B (receipt option LINK, GET, or POST, ssl_dcc_popup_url defined)
A response data stream is returned to the ssl_dcc_popup_url as a GET request.
Example

```
http://www.merchanturlhere.com/dccpopupurl.com?id= PBczyFq67o8Vfzx9LdPyPKS
&ssl_txn_currency_code=CAD&ssl_conversion_rate=1.2345&ssl_amount=1.00&ssl_cardholder_amount=
1.23&dccoption=%7B%28option+label%3D%27Please+charge+my+purchase+in+my+home+currency%2
7%29%3DY%3B%28option+label%3D%27Do+not+charge+me+in+my+home+currency%3B+charge+my
+purchase+in+US+dollars%27%29%3DN%7D
```

Sending Device: Data stream sent to process.do
Example

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
 <input type="hidden" name="ID " value="PBczyFq67o8Vfzx9LdPyPKS ">
 <input type="hidden" name="dccoption" value="Y">
 <input type="submit" value="Continue">
</form>
```

process.do response will be as in earlier examples.

END OPTION B

process.do response:

OPTION C (receipt option REDG, No ssl_dcc_popup_url defined)
A response data stream is returned to the requesting device.
Example

```
id=PBczyFq67o8Vfzx9LdPyPKS
&ssl_txn_currency_code=CAD&ssl_conversion_rate=1.2345&ssl_amount=1.00&ssl_cardholder_amount=
1.23&dccoption=%7B%28option+label%3D%27Please+charge+my+purchase+in+my+home+currency%2
```

7%29%3DY%3B%28option+label%3D%27Do+not+charge+me+in+my+home+currency%3B+charge+my+purchase+in+US+dollars%27%29%3DN%7D

Sending Device: Data stream sent to process.do
Example

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
  <input type="hidden" name="ID " value="PBczyFq67o8Vfzx9LdPyPKS">
  <input type="hidden" name="dccoption" value="Y">
  <input type="submit" value="Continue">
</form>
```

process.do response will be as in earlier examples.

END OPTION C

process.do response:

OPTION D (receipt option REDG, ssl_dcc_popup_url defined)
A response data stream is returned to the ssl_dcc_popup_url as a GET request.
Example

http://www.merchanturlhere.com/dccpopupurl.com?id=PBczyFq67o8Vfzx9LdPyPKS&ssl_txn_currency_code=CAD&ssl_conversion_rate=1.2345&ssl_amount=1.00&ssl_cardholder_amount=1.23&dccoption=%7B%28option+label%3D%27Please+charge+my+purchase+in+my+home+currency%27%29%3DY%3B%28option+label%3D%27Do+not+charge+me+in+my+home+currency%3B+charge+my+purchase+in+US+dollars%27%29%3DN%7D

Sending Device: Data stream sent to process.do
Example

```
<form action="https://www.myvirtualmerchant.com/VirtualMerchant/process.do" method="POST">
  <input type="hidden" name="ID " value="PBczyFq67o8Vfzx9LdPyPKS">
  <input type="hidden" name="dccoption" value="Y">
  <input type="submit" value="Continue">
</form>
```

process.do response will be as in earlier examples.

END OPTION D

Best Practices

Developers and merchant administrators may find the information presented here valuable when writing and configuring applications and Websites that will interface with Virtual Merchant. Merchant Administrators may find this information valuable as well. These best practices focus on ways to increase security and reduce the chance of fraudulent activity.

- **HTTP Referrer –** Setting up the use of HTTP Referrers in the admin site tells Virtual Merchant to only accept transactions from a pre-approved list of Websites. While requiring more work to implement, this action will prevent fraudulent users from submitting transactions from their Website, claiming to be you.

- **Server Side Code –** Your users can read HTML source code from your Web pages when they are downloaded to the Web browser. Although our simple examples in the document show this as a method for passing data to Virtual Merchant, we do not recommend this for your production Website. All sensitive merchant data, including transaction amount and your Virtual Merchant credentials, should be placed in server side code, rather than of just hidden value fields on an HTML form. This will reduce the ability for malicious users to edit and use this data for their own fraudulent purposes.

- **Auto Pend –** We recommend that you use the Auto-Pend feature for any account that is set to Auto-Settle. This gives you the chance to review each transaction before it becomes finalized. This will help you avoid settling fraudulent transactions or transactions that you are unable to fulfill.

- **Merchant Admin –** The Merchant Admin account has full rights and access to each terminal in your system. We recommend that you use this account sparingly. We suggest that you create one or more separate accounts to manage day-to-day activity, including but not limited to: processing transactions from your Website, processing Virtual Terminal transactions, reviewing transactions and settling transactions. We recommend you do this even in the case of an Account with only one terminal.

- **Password Security –** Do not set your password to be the same value (or a similar value) as any other data associated with your Virtual Merchant account. This includes your Virtual Merchant PIN used for submitting transactions to process.do. This PIN is not designed as a security feature. It is only used to ensure that transactions sent into Virtual Merchant are assigned to the correct Account, User and Terminal. Unlike the passwords, the PIN is not stored as encrypted data in our database. Your password is a highly confidential piece of data and is treated as such. Our administrators do not have access to your password data. You should make all of your Accounts' passwords as difficult to guess as possible.

- **Settings in Admin site –** We recommend that whenever possible you set Terminal options in the Administrative site, instead of setting equivalent parameters in code on your Web page. This will make it easier to maintain and will reduce the amount of data that is passed across the Internet with each of your transactions.

- **Business Rules** – This section allows you to set up a list of authorization rules to perform some action on the transaction, such as mark the record for review or send an email. This is only available for Terminal-based processing.

Features

The information here is targeted at Developers writing applications and Websites that will interface with Virtual Merchant, as well as Merchant Administrators who manage individual Virtual Merchant accounts.

- **HTTP Referrer –** The HTTP Referrer feature is used to tell Virtual Merchant to accept only transactions from a pre-approved list of Websites. This will prevent fraudulent users from submitting transactions from their Website, claiming to be you.

- **Separate Decline Receipt Approved Receipt Options –** The split Receipt feature allows Virtual Merchant to handle Approved and Declined transactions differently. And more importantly, it allows you to tell Virtual Merchant about these differences. You can specify different URLs for your own custom receipts; allow Virtual Merchant to pass different transaction information back to you and much more.

- **Multiple Terminals per VID –** Virtual Merchant accounts are now capable of having multiple terminals assigned to them. This allows you keep all of your Virtual Merchant transactions together. Whether you have multiple Websites, a MO/TO call center, or multiple retail store locations, you can now manage all of your accounts from a single Virtual Merchant account.

- **MO/TO and Retail Terminals –** Virtual Merchant now supports terminals for the MO/TO and Retail market place, as well as the Internet. This allows you to take advantage of better Interchange rates for transactions in those Market Segments.

- **User Hierarchy including user Rights, and Terminal association –** The new User hierarchy allows you to assign different user ID's to each person using your account on Virtual Merchant. This allows to you assign different rights and functions to specific people in your organization. This feature also will allow you track which people are making new sales.

- **Disabling HTTP Transactions –** Terminals can now be set to deny all HTTP transactions from process.do. This way if a terminal is only accessed from the Virtual Merchant Virtual Terminal, Virtual Merchant will know not to accept transactions for this terminal sent into process.do from Websites on the Internet.

- **Advanced Headers/Footers –** Virtual Merchant now allows you to use a special Virtual Merchant Markup Language to present custom headers and footers to your customers on e-mails, the payment form and the receipt page. True HTML tags are not allowed for security reasons, but the text and tags you enter will be translated into HTML to allow you to present more advanced content to your customers.

- **Card Swipe –** The Card Swipe feature allows Virtual Merchant to accept card data from a Magnetic Card Reader input device attached to your computer. This allows you to process face to face retail transactions using the Virtual Terminal. Along with the added convenience to both you and your customers, these types of transactions result in the lowest Interchange Rates possible!

- **Credit Card Processing –** Supports processing for all major types of credit cards.

- **Debit Card Processing –** Supports processing for Debit/ATM PIN bases transactions. Terminals set up for Internet and MOTO transactions can be configured to accept Pinless Debit.

- **Electronic Gift Card Processing –** Allows you to extend more payment choices to your customers while adding value to your business. Gift cards enable you to expand your market, increase revenue, and build customer loyalty, all with electronic efficiency.

- **Dynamic Currency Conversion (DCC) –** If the merchant has the "DCC Processing" option enabled, then the application checks the currency listed for the BIN. If it is different than the merchant currency, then it displays the information for the conversion on the input screen. The merchant has the option to continue the transaction with the pre-converted amount, or to proceed without the conversion based upon the consumer's choice.

- **Serial Receipt Printer –** To add to the capabilities to handle Retail Terminals, printed receipts for transactions are now supported. This includes the ability to interface with a Serial Receipt Printer with options to specify paper type, headers and footers, as well as the ability to print out your own custom fields.

- **PINs –** Personal Identification Numbers (PINs) can be up to six characters in length and contain full alphanumeric data.

- **Error Configuration –** You can customize some error messages that the system displays. This allows you to specify contact information, helpful tips, multi-language instructions, or other information that you didn't have the option to present to your customer before.

Appendices

Appendix 1: Input values

FIELD_NAME	LENGTH	DEFAULT	REQ	DESCRIPTION
ssl_test_mode	5	FALSE	N	Optional, when set to TRUE, transactions will not be forwarded to the credit card processor, but instead will always return an "APPROVED" result.
ssl_transaction_type	20		Y	**Credit Card Transaction Types** Sale(CCSALE) Auth Only(CCAUTHONLY) Credit(CCCREDIT) Force(CCFORCE) Balance Inquiry(CCBALINQUIRY) **EGC Transaction Types** Activation(EGCACTIVATION) Sale / Redemption(EGCSALE) Card Refund(EGCCARDREFUND) Replenishment / Reload(EGCRELOAD) Card Balance Inquiry(EGCBALINQUIRY) Credit(EGCCREDIT) **PIN Less Debit Transaction Types** PINLess Debit Purchase(PLDPURCHASE)
ssl_merchant_id	15		Y	Virtual Merchant ID
ssl_pin	6		Y	PIN
ssl_user_id	15	ssl_merchant_id	Y	USER ID
ssl_amount	13		Y	Transaction Amount
ssl_salestax	10		N	Sales Tax
ssl_card_number	19		Y	Card Number
ssl_exp_date	4		Y	Card Expiry Date
ssl_cvv2cvc2_indicator	11		N	CVV2 Indicator 0= Bypassed 1=present, , 2=Illegible, and 9=Not Present
ssl_cvv2cvc2	4		N	CVV2 value
ssl_description	255		N	Transaction Description
ssl_invoice_number	25		N	Invoice Number
ssl_customer_code	25		N	Customer Code
ssl_company	50		N	Customer's company name
ssl_first_name	20		N	Customer's first name
ssl_last_name	30		N	Customer's last name
ssl_avs_address	20		N	Customer's address used to process AVS
ssl_address2	30		N	Customer's address line 2
ssl_city	30		N	Customer's City
ssl_state	30		N	Customer's State
ssl_avs_zip	9		N	Customer's zip code used to process AVS
ssl_country	50		N	Customer's Country

ssl_phone	20		N	Customer's Phone Number
ssl_email	100		N	Customer's Email Address
ssl_ship_to_company	50		N	Ship To Company Name
ssl_ship_to_first_name	20		N	Ship To First Name
ssl_ship_to_last_name	30		N	Ship To Last Name
ssl_ship_to_address1	30		N	Ship To Address Line 1
ssl_ship_to_address2	30		N	Ship To Address Line 2
ssl_ship_to_city	30		N	Ship To City
ssl_ship_to_state	30		N	Ship To State
ssl_ship_to_zip	10		N	Ship To Zip
ssl_ship_to_country	50		N	Ship To Country
ssl_ship_to_phone	20		N	Ship To Phone Number
ssl_email_header	255		N	Customer order confirmation email header; If present, overrides values for ssl_email_apprvl_header_html and ssl_email_decl_header_html
ssl_email_apprvl_header_html	255	Set in admin	N	Customer order confirmation email header for approvals
ssl_email_decl_header_html	255	Set in admin	N	Customer order confirmation email header for declines
ssl_email_footer	255	Set in admin	N	Customer order confirmation email footer; If present, overrides values for ssl_email_apprvl_footer_html and ssl_email_decl_footer_html
ssl_email_apprvl_footer_html	255		N	Customer order confirmation email footer for approvals
ssl_email_decl_footer_html	255		N	Customer order confirmation email footer for declines
ssl_do_customer_email	1	set in admin (TRUE | FALSE)	N	
ssl_do_merchant_email	1	Set in admin (TRUE | FALSE)	N	
ssl_merchant_email	100	Set in Admin	N	Merchant's Email Address
ssl_header_color	20	Set in Admin	N	Any HTML Color Value
ssl_text_color	20	Set in Admin	N	Any HTML Color Value
ssl_background_color	20	Set in Admin	N	Any HTML Color Value
ssl_table_color	20	Set in Admin	N	Any HTML Color Value
ssl_link_color	20	Set in Admin	N	Any HTML Color Value
ssl_show_form	5	TRUE	N	When set to FALSE Virtual Merchant will not present the payment form but process the transaction directly.
ssl_header_html	255	Set in Admin	N	payment form header ignored when ssl_show_form=FALSE

ssl_footer_html	255	Set in Admin	N	payment form footer ignored when ssl_show_form=FALSE
ssl_result_format	5	HTML	N	When set to ASCII Virtual Merchant will generate a plain text key-value document.
ssl_receipt_header_html	255	Set in Admin	N	Receipt form header ignored when ssl_result_format=ASCII
ssl_receipt_apprvl_header_html	255	Set in Admin	N	Receipt form header for Approved Transaction. Ignored when ssl_result_format=ASCII
ssl_receipt_decl_header_html	255	Set in Admin	N	Receipt form header for Declined Transaction. Ignored when ssl_result_format=ASCII
ssl_receipt_footer_html	255	Set in Admin	N	Receipt form footer Ignored when ssl_result_format=ASCII
ssl_receipt_apprvl_footer_html	255	Set in Admin	N	Receipt form footer for Approved Transaction. Ignored when ssl_result_format=ASCII
ssl_receipt_decl_footer_html	255	Set in Admin	N	Receipt form footer for Declined Transaction. Ignored when ssl_result_format=ASCII
ssl_receipt_link_method	4	Set in admin (LINK | GET | POST | REDG)	N	REDG = No receipt displayed. data redirected to ssl_receipt_link_url LINK = Receipt displayed. Link provided to return to ssl_receipt_link_url GET = Receipt displayed. Button provided to send Get data to ssl_receipt_link_url POST = Receipt displayed. Button provided to send Post data to ssl_receipt_link_url LINK, GET, POST ignored when ssl_result_format=ASCII If present, overwrites ssl_receipt_apprvl_method and ssl_receipt_decl_method
ssl_receipt_apprvl_method	4	Set in admin (LINK | GET | POST | REDG)	N	REDG = No receipt displayed. data redirected to ssl_receipt_link_url LINK = Receipt displayed. Link provided to return to ssl_receipt_link_url GET = Receipt displayed. Button provided to send Get data to ssl_receipt_link_url POST = Receipt displayed. Button provided to send Post data to ssl_receipt_link_url LINK, GET, POST ignored when ssl_result_format=ASCII If present, overwrites ssl_receipt_apprvl_method and ssl_receipt_link_method

ssl_receipt_decl_method	4	Set in admin (LINK | GET | POST | REDG)	N	REDG = No receipt displayed. data redirected to ssl_receipt_link_url LINK = Receipt displayed. Link provided to return to ssl_receipt_link_url GET = Receipt displayed. Button provided to send Get data to ssl_receipt_link_url POST = Receipt displayed. Button provided to send Post data to ssl_receipt_link_url LINK, GET, POST ignored when ssl_result_format=ASCII If present, overwrites ssl_receipt_apprvl_method and ssl_receipt_link_method
ssl_receipt_link_url	255	set in admin	N	Target of the Redirect or the link generated at the bottom of the Virtual Merchant drawn receipt. Ignored when ssl_result_format=ASCII and ssl_receipt_link_method = LINK, GET, or POST. If present, overwrites ssl_receipt_apprvl_post_url, ssl_receipt_decl_post_url, ssl_receipt_apprvl_get_url, and ssl_receipt_decl_get_url
ssl_receipt_apprvl_post_url	255	set in admin	N	Target of the link generated at the bottom of the receipt for an approval using the "POST" method. Ignored when ssl_result_format=ASCII
ssl_receipt_decl_post_url	255	set in admin	N	Target of the link generated at the bottom of the receipt for a declined transaction using the "POST" method. Ignored when ssl_result_format=ASCII
ssl_receipt_apprvl_get_url	255	set in admin	N	Target of the link generated at the bottom of the receipt for an approval using the "GET" method, or the Target of the redirect for an approval using the "REDG" method. Ignored when ssl_result_format=ASCII and ssl_receipt_link_method = LINK, GET, or POST.
ssl_receipt_decl_get_url	255	set in admin	N	Target of the link generated at the bottom of the receipt for a declined transaction using the "GET" method, or the Target of the redirect for a declined transaction using the "REDG" method. Ignored when ssl_result_format=ASCII and ssl_receipt_link_method = LINK, GET, or POST.
ssl_error_url	255	set in admin	N	If present redirect the error with ssl_error_code, ssl_error_name, and ssl_error_message to the URL.

ssl_receipt_link_text	40	set in admin	N	Text in the link / on the submit button generated at the bottom of the receipt page. Ignored when ssl_result_format=ASCII; If present, overwrites ssl_receipt_apprvl_text and ssl_receipt_decl_text
ssl_receipt_apprvl_text	40	set in admin	N	Text that appears on the receipts of approved transactions
ssl_receipt_decl_text	40	set in admin	N	Text that appears on the receipts of declined transactions
ssl_result	4	0	N	Result code for the transaction: a result of 0 indicates an approval; any other result means that the transaction was not approved.
ssl_result_message	4	APPROVED	N	Result message for the transaction: a result of "APPROVED" indicates an approval; any other result means that the transaction was not approved.
ssl_txn_id	36		N	Unique transaction identifier
ssl_approval_code	6		N	Return code generated by credit card processor
ssl_cvv2_response	1		N	Return code generated by service. See Appendix 3: CVV2/CVC Response Codes
ssl_avs_response	1		N	Return code generated by service. See Appendix 2: AVS Response Codes
PLD (PinLess Debit)				
ssl_customer_number	25		y	Customer Number
ssl_account_type	8	0	y	Account Type (savings 1, checking 0)
ssl_surcharge_amount	5	set in admin	N	Value cannot be submitted
EGC (Electronic Gift Card)				
ssl_egc_tender_type	6		Y	0=cash 1=credit 2=debit 3=check
DCC (Dynamic Currency Conversion)				
ssl_dcc_popup_url	255		N	If present redirect the dcc decision page to the specified URL. Only used for ssl_result_format=ASCII or XML

ID	25		Y	Session ID used for DCC Decision page.
ssl_txn_currency_code	3			Code indicating cardholder currency for the DCC Decision page
ssl_conversion_rate	8			Currency multiplier from merchant currency to cardholder currency.
ssl_cardholder_amount	12			Transaction amount in the cardholder's currency
Dccoption	1		Y	Returned value for DCC Decision. Y = Yes proceed with customer currency. N= No proceed in merchant currency.
• Please charge my purchase in my home currency = Y				
• Do not charge me in my home currency; charge my purchase in US dollars = N				
XML Fields				
<txn></txn>			Y	Qualifier used to wrap all XML transactional data
Xmldata			Y	All XML data submitted should be attached to this field in format …/processxml.do?xmldata=(XML data here) All XML data is also returned in this field in the same format when a redirect URL is used.

Appendix 2: Response Codes

Authorization Response Codes

This is a list of the values that may be returned during an authorization request.

Authorization Response Codes		
Code	**Message**	**Definition**
AA	APPROVAL	Approved
AA	APPROVAL PARTIAL	Approved for a Partial Amount
ND	AMOUNT ERROR	Tran Amount Error
N7	DECLINE CVV2	Do Not Honor
ND	EXPIRED CARD	Expired Card
ND	INVLD TERM ID 1	Invalid Merchant Number
ND	SEQ ERR PLS CALL	Call for Assistance
ND	AMT OVER SVC LMT	Amount is more than established service limit
ND	APPL TYPE ERROR	Call for Assistance
NR	CALL AUTH. CENTER	Refer to Issuer
NR	CALL REF.; 999999	Refer to Issuer
ND	CANNOT CONVERT	Check is ok, but cannot be converted. Do Not Honor
N7	DECLINE CVV2	Declined due to CVV2 mismatch \ failure
ND	DECLINED	Do Not Honor
ND	DECLINED T4	Do Not Honor. Failed negative check, unpaid items
ND	DECLINED-HELP 9999	System Error
ND	DUP CHECK NBR	Duplicate Check Number
ND	INCORRECT PIN	Invalid PIN
ND	INVALID CARD	Invalid Card
ND	INVALID CAVV	Invalid Cardholder Authentication Verification Value
ND	INVALID TERM ID	Invalid Terminal ID
ND	INVLD R/T NBR	Invalid Routing/Transit Number
ND	INVLD TERM ID 2	Invalid SE Number
ND	INVLD VOID DATA	Invalid Data Submitted for Void Transaction
ND	MAX MONTHLY VOL	The maximum monthly volume has been reached
ND	MICR ERROR	MICR Read Error
ND	MUST SETTLE MMDD	Must settle, open batch is over 7 days old. **Note:** Batch will be Auto Settled after 10 days
ND	NETWORK ERROR	General System Error
NC	PICK UP CARD	Pick up card

ND	PLEASE RETRY	Please Retry/ Reenter Transaction
ND	RECORD NOT FOUND	Record not on NOVA Network
ND	REQ. EXCEEDS BAL.	Req. exceeds balance
ND	SERV NOT ALLOWED	Invalid request
ND	TOO MANY CHECKS	Too Many Checks (Over Limit)
AA	APPROVAL	Approved
AA	APPROVAL PARTIAL	Approved for a Partial Amount
ND	AMOUNT ERROR	Tran Amount Error
N7	DECLINE CVV2	Do Not Honor

Electronic Gift Card (EGC) Response Codes

This is a list of the values that may be returned during an EGC authorization request.

Authorization Response Codes		
Code	Message	Definition
AA	APPROVAL	Approved
01	DECLINED-HELP 9999	Host Busy
07	MAX REACHED	Cannot load the amount specified
13	DUPLICATE TRAN	Duplicate transaction
02	INVALID CARD	Invalid Card
03	INVALID TERM ID	Invalid Terminal ID
04	AMOUNT ERROR	Tran Amount Error
05	ALREADY ACTIVE	Card already active
06	REQ. EXCEEDS BAL.	Request exceeds balance
08	NON RELOADABLE	The card cannot be reloaded
09	TRAN NOT ALLOWED	Transaction type not allowed
10	INVLD TRAN TYPE	Transaction type not on server
11	EXPIRED CARD	Expired card or bad expiration date
12	CARD NOT ACTIVE	The Gift Card is not activated
14	SEQ ERR PLS CALL	Call for Assistance
15	SEQ ERR PLS CALL	Sequence does not match previous response
16	INVALID BATCH ID	Batch ID is not on the server
17	INVALID TENDER	Tender types is not on the server
99	DECLINED-HELP 9999	General System Error
ND	SERV NOT ALLOWED	Invalid request
ND	INVLD TERM ID 1	Invalid Merchant Number
ND	SEQ ERR PLS CALL	Call for Assistance
ND	APPL TYPE ERROR	Call for Assistance
AA	APPROVAL	Approved

AVS Response Codes

An **AVS Response Code** is returned in Authorization Response Message when AVS information is present in the transaction authorization request.

AVS Response Codes	
Code	**Definition**
A	Address matches - Zip Code does not match.
B	Street address match, Postal code in wrong format. (International issuer)
C	Street address and postal code in wrong formats
D	Street address and postal code match (international issuer)
E	AVS Error
G	Service not supported by non-US issuer
I	Address information not verified by international issuer.
M	Street Address and Postal code match (international issuer)
N	No Match on Address (Street) or Zip
O	No Response sent
P	Postal codes match, Street address not verified due to incompatible formats.
R	Retry, System unavailable or Timed out
S	Service not supported by issuer
U	Address information is unavailable
W	9 digit Zip matches, Address (Street) does not match.
X	Exact AVS Match
Y	Address (Street) and 5-digit Zip match.
Z	5 digit Zip matches, Address (Street) does not match.

CVV2/CVC Response Codes

The **CVV2 Response Codes** are returned in an Authorization Response Message when the CVV2 data is present in the transaction authorization request.

CVV2 Response Codes	
Code	**Definition**
M	CVV2 Match
N	CVV2 No match
P	Not Processed
S	Issuer indicates that CVV2 data should be present on the card, but the merchant has indicated that the CVV2 data is not resent on the card

U	Issuer has not certified for CVV2 or Issuer has not provided Visa with the CVV2 encryption Keys.

Appendix 3: Virtual Merchant Error Numbers and Messages

A **Virtual Merchant Error Number and Error Message** is returned when the transaction fails to be authorized. This could be the result of a data or system error, or if the transaction is "Declined." For backwards compatibility, the non-unique v1 error number is currently returned.

Number	Error Name	Default Message	Custom Message
3000	Gateway not responding	Error, no response.	Error, no response.
3001	Gateway generated error	#.	#.
3002	Adapter generated error	#.	#.
4000	VID Not Supplied	The VirtualMerchant ID was not supplied in the authorization request.	The VirtualMerchant ID was not supplied in the authorization request.
4001	VID, UID and PIN Invalid	The VirtualMerchant ID, User ID and/or PIN supplied in the authorization request is invalid.	The VirtualMerchant ID, User ID and/or PIN supplied in the authorization request is invalid.
4002	HTTP Trans Not Allowed	HTTP POST transactions are not allowed for this account.	HTTP POST transactions are not allowed for this account.
4003	HTTP Referrer Invalid	HTTP POST transactions are not allowed for this HTTP Referrer.	HTTP POST transactions are not allowed for this HTTP Referrer.
4005	E-mail Address Invalid	The E-mail Address supplied in the authorization request appears to be invalid.	The E-mail Address supplied in the authorization request appears to be invalid.
4006	CVV2 Not Requested With Data	The CVV2 indicator was not identified in the authorization request.	The CVV2 indicator was not identified in the authorization request.
4007	CVV2 Requested But No Data	CVV2 check cannot be performed as no data was supplied in the authorization request.	CVV2 check cannot be performed as no data was supplied in the authorization request.
4009	Required Field Not Supplied	A required field was not supplied in the authorization request.	A required field was not supplied in the authorization request.
4010	Invalid Transaction Type	An invalid Transaction Type was supplied in the authorization request.	An invalid Transaction Type was supplied in the authorization request.
4011	Receipt URL Missing	The Receipt URL supplied in the authorization request appears to be blank or invalid.	The Receipt URL supplied in the authorization request appears to be blank or invalid.
4012	VID/UID Invalid	The VirtualMerchant ID and/or User ID supplied in the authorization request is invalid.	The VirtualMerchant ID and/or User ID supplied in the authorization request is invalid.
4013	PIN Not Supplied	The PIN was not supplied in the authorization request.	The PIN was not supplied in the authorization request.
4014	Not Permitted	This account is not permitted to process this transaction type.	This account is not permitted to process this transaction type

4015	PIN Invalid	The PIN supplied in the authorization request is invalid.	The PIN supplied in the authorization request is invalid.
4016	Permission Denied	This account does not have permission to process # transactions.	This account does not have permission to process # transactions.
5000	Credit Card Number Invalid	The Credit Card Number supplied in the authorization request appears to be invalid.	The Credit Card Number supplied in the authorization request appears to be invalid.
5001	Exp Date Invalid	The Credit Card Expiration Date supplied in the authorization request appears to be invalid.	The Credit Card Expiration Date supplied in the authorization request appears to be invalid.
5002	Amount Invalid	The amount supplied in the authorization request appears to be invalid.	The amount supplied in the authorization request appears to be invalid.
5003	Approval Code / No Force	A FORCE Approval Code was supplied for this transaction, however the transaction type is not FORCE.	A FORCE Approval Code was supplied for this transaction, however the transaction type is not FORCE.
5004	Invalid Approval Code	The FORCE Approval Code supplied in the authorization request appears to be invalid or blank. The FORCE Approval Code must be 6 or less alphanumeric characters.	The FORCE Approval Code supplied in the authorization request appears to be invalid or blank. The FORCE Approval Code must be 6 or less alphanumeric characters.
5005	Field Character Limit Exceeded	The value for the # field is too long. # characters (maximum) are allowed. Your entry contains # characters. If you entered the value for this field, use the browser BACK button to return to the order form and modify the field value accordingly. Otherwise, contact Customer Service at #.	The value for the # field is too long. # characters (maximum) are allowed. Your entry contains # characters. If you entered the value for this field, use the browser BACK button to return to the order form and modify the field value accordingly. Otherwise, contact Customer Service at #.
5006	Refund Amount Exceeds Limit	The refund amount for this transaction ($#) may not exceed $#.	The refund amount for this transaction ($#) may not exceed $#.
5007	Sales Tax Invalid	The Sales Tax supplied in the authorization request appears to be invalid.	The Sales Tax supplied in the authorization request appears to be invalid.
5008	Invalid Account Type	This PIN Less Debit Transaction contains invalid account type. Account type can be checking or saving	This PIN Less Debit Transaction contains invalid account type. Account type can be checking or saving
5009	Invalid Surcharge Amount	Invalid Surcharge amount for the PIN less debit transaction	Invalid Surcharge amount for the PIN less debit transaction
5010	Invalid EGC Transaction type	An invalid EGC Transaction	An invalid EGC Transaction

		type has been supplied with this request	type has been supplied with this request
5011	Invalid EGC Tender Type	An invalid EGC Tender type has been supplied with this request	An invalid EGC Tender type has been supplied with this request
6001	Manual Transaction Declined	The transaction request was unable to be completed:	The transaction request was unable to be completed:
6002	Declined: Invalid Card	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6003	Declined: Pick up Card	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6004	Declined: Amount Error	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6005	Declined: Appl. Type Error	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6006	Declined	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6007	Declined: Help	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6008	Declined: Req. Exceeds Bal.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6009	Declined: Expired Card	This transaction request has	This transaction request has

		not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6010	Declined: Incorrect PIN	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6011	Declined: Invalid Term ID	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6012	Declined: Invalid Term ID 1	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6013	Declined: Invalid Term ID 2	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6014	Declined: Invalid Void Data	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6015	Declined: Must Settle MMDD	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6016	Declined: Not On File	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6017	Declined: Record Not Found	This transaction request has not been approved. You may elect to use another form of	This transaction request has not been approved. You may elect to use another form of

		payment to complete this transaction or contact customer service for additional options.	payment to complete this transaction or contact customer service for additional options.
6018	Declined: Serv Not Allowed	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6019	Declined: Seq Err Pls Call	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6020	Declined: Call Auth Center	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6021	Declined: Call Ref.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6022	Declined: CVV2	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6023	Declined: Please RetryXXXX	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.	This transaction request has not been approved. You may elect to use another form of payment to complete this transaction or contact customer service for additional options.
6024	Card Already Active	The Gift Card is already active	The Gift Card is already active
6025	Request Exceeds Balance	The transaction amount exceeds the Gift Card balance amount	The transaction amount exceeds the Gift Card balance amount
6026	Cannot Load The Amount Specified	Cannot Load The Amount Specified	Cannot Load The Amount Specified
6027	Card Not Activated	The Gift Card Is Not Activated	The Gift Card Is Not Activated
6028	Card Cannot Be Reloaded	The Gift Card Cannot Be Reloaded	The Gift Card Cannot Be Reloaded

6029	Declined: Invalid Reg Key	Invalid Reg Key	Invalid Reg Key
6030	Declined: Invalid Packet	Invalid Packet	Invalid Packet
6031	Declined: Invalid LRC	Invalid LRC	Invalid LRC
6032	Declined: Invalid Response	Invalid Response	Invalid Response
6033	Declined: Invalid LRC in Response	Invalid LRC in Response	Invalid LRC in Response
6034	Declined: Invalid Record Number in Response	Invalid Record Number in Response	Invalid Record Number in Response
6038	System is Temporarily Unavailable	It appears that the system is temporarily unavailable. Please try your transaction again in a few minutes or contact the merchant you are trying to order from for further assistance. We apologize for this inconvenience.	It appears that the system is temporarily unavailable. Please try your transaction again in a few minutes or contact the merchant you are trying to order from for further assistance. We apologize for this inconvenience.

Appendix 4: ProcessXML.do

Virtual Merchant provides a means to integrate in XML format. The URL
https://www.myvirtualmerchant.com/VirtualMerchant/processxml.do is provided to receive data only in
XML format. All XML data should be sent to this URL with the key xmldata. An example is provided
below. As with integrations with process.do URL, the system can redirect to a supplied URL when an
error occurs or forward approval/decline data to a specified address.

Example 1

In this example the XML data provides everything needed to perform a basic transaction and receive a
response to your Website.

Data string sent to processxml.do
…/VirtualMerchant/processxml.do?xmldata=<txn><ssl_merchant_ID>123456</ssl_merchant_ID><ssl_us
er_id>123456</ssl_user_id><ssl_pin>V6NJ3A</ssl_pin><ssl_transaction_type>ccsale</ssl_transaction_t
ype><ssl_card_number>1111111111111111</ssl_card_number><ssl_exp_date>1210</ssl_exp_date><s
sl_amount>2.34</ssl_amount><ssl_salestax>0.00</ssl_salestax><ssl_cvv2cvc2_indicator>1</ssl_cvv2cv
c2_indicator><ssl_cvv2cvc2>321</ssl_cvv2cvc2><ssl_invoice_number>1234</ssl_invoice_number><ssl_
customer_code>1111</ssl_customer_code><ssl_first_name>customer</ssl_first_name><ssl_last_name>
name</ssl_last_name><ssl_avs_address>1234 main st.</ssl_avs_address><ssl_address2>apt
b</ssl_address2><ssl_city>any
town</ssl_city><ssl_state>ST</ssl_state><ssl_avs_zip>55555</ssl_avs_zip><ssl_phone>555-555-
5555</ssl_phone><ssl_email>customer@email.com</ssl_email></txn>

Response
<?xml version="1.0" encoding="UTF-8" ?>
- <txn>
  <ssl_result>0</ssl_result>
  <ssl_result_message>APPROVED</ssl_result_message>
  <ssl_card_number>11********1111</ssl_card_number>
  <ssl_exp_date>1208</ssl_exp_date>
  <ssl_amount>2.34</ssl_amount>
.
. (other xml fields)
.
  <ssl_txn_id>14039F9DA-1428-B5BB-508B-634707DC3413</ssl_txn_id>
  <ssl_approval_code>N35032</ssl_approval_code>
  <ssl_cvv2_response>P</ssl_cvv2_response>
  <ssl_avs_response>X</ssl_avs_response>
  <ssl_account_balance>0.00</ssl_account_balance>
  <ssl_txn_time>10/26/2006 12:35:03 PM</ssl_txn_time>
  </txn>

Example 2

This example expands on the basic example by supplying a redirecting URL and Error URL so that any errors or responses can be forwarded to a different address from the requestor.

Data string sent to processxml.do
…/VirtualMerchant/processxml.do?xmldata=\<txn>\<ssl_merchant_ID>123456\</ssl_merchant_ID>\<ssl_user_id>123456\</ssl_user_id>\<ssl_pin>V6NJ3A\</ssl_pin>\<ssl_transaction_type>ccsale\</ssl_transaction_type>\<ssl_card_number>1111111111111111\</ssl_card_number>\<ssl_exp_date>1210\</ssl_exp_date>\<ssl_amount>2.34\</ssl_amount>\<ssl_salestax>0.00\</ssl_salestax>\<ssl_cvv2cvc2_indicator>1\</ssl_cvv2cvc2_indicator>\<ssl_cvv2cvc2>321\</ssl_cvv2cvc2>\<ssl_invoice_number>1234\</ssl_invoice_number>\<ssl_customer_code>1111\</ssl_customer_code>\<ssl_first_name>customer\</ssl_first_name>\<ssl_last_name>name\</ssl_last_name>\<ssl_avs_address>1234 main st.\</ssl_avs_address>\<ssl_address2>apt b\</ssl_address2>\<ssl_city>any town\</ssl_city>\<ssl_state>ST\</ssl_state>\<ssl_avs_zip>55555\</ssl_avs_zip>\<ssl_phone>555-555-5555\</ssl_phone>\<ssl_email>customer@email.com\</ssl_email>\<ssl_receipt_link_method>REDG\</ssl_receipt_link_method>\<ssl_receipt_link_url>https://www.merchanturl.com/receipt.html\</ssl_receipt_link_url>\<ssl_error_url>https://www.merchanturl.com/errorpage.html\</ssl_error_url>\</txn>

Response forwarded to the receipt link URL as attached to the xmldata field

https://merchanturl/receiptpage.html?xmldata=\<txn>\<ssl_result>0\</ssl_result>\<ssl_result_message>APPROVED\</ssl_result_message>\<ssl_card_number>11********1111\</ssl_card_number>\<ssl_exp_date>1208\</ssl_exp_date>\<ssl_amount>2.34\</ssl_amount>... (other xml fields)
  \<ssl_txn_id>14039F9DA-1428-B5BB-508B-634707DC3413\</ssl_txn_id>\<ssl_approval_code>N35032\</ssl_approval_code>\<ssl_cvv2_response>P\</ssl_cvv2_response>\<ssl_avs_response>X\</ssl_avs_response>\<ssl_account_balance>0.00\</ssl_account_balance>\<ssl_txn_time>10/26/2006 12:35:03 PM\</ssl_txn_time>\</txn>

If an error occurred that data would be submitted in this format.

https://www.merchanturl.com/errorpage.html?xmldata=\<txn>\<errorCode>ErrorNumber\</errorCode>\<errorName>ErrorName\</errorName>\<errorMessage>ErrorMessage\</errorMessage>\</txn>

Example 3

This example provides a detailed transaction using the approval/decline URLs.

…/VirtualMerchant/processxml.do?xmldata=<txn><ssl_merchant_ID>123456</ssl_merchant_ID><ssl_user_id>123456</ssl_user_id><ssl_pin>V6NJ3A</ssl_pin><ssl_transaction_type>ccsale</ssl_transaction_type><ssl_card_number>1111111111111111</ssl_card_number><ssl_exp_date>1210</ssl_exp_date><ssl_amount>2.34</ssl_amount><ssl_salestax>0.00</ssl_salestax><ssl_cvv2cvc2_indicator>1</ssl_cvv2cvc2_indicator><ssl_cvv2cvc2>321</ssl_cvv2cvc2><ssl_invoice_number>1234</ssl_invoice_number><ssl_customer_code>1111</ssl_customer_code><ssl_first_name>customer</ssl_first_name><ssl_last_name>name</ssl_last_name><ssl_avs_address>1234 main st.</ssl_avs_address><ssl_address2>apt b</ssl_address2><ssl_city>any town</ssl_city><ssl_state>ST</ssl_state><ssl_avs_zip>55555</ssl_avs_zip><ssl_phone>555-555-5555</ssl_phone><ssl_email>customer@email.com</ssl_email><ssl_receipt_apprvl_method>REDG</ssl_receipt_apprvl_method><ssl_receipt_decl_method>REDG</ssl_receipt_decl_method><ssl_receipt_apprvl_get_url>https://www.merchanturl.com/approved.html</ssl_receipt_apprvl_get_url><ssl_receipt_decl_get_url>https://www.merchanturl.com/declined.html</ssl_receipt_decl_get_url><ssl_error_url>https://merchanturl.com/errorpage.html</ssl_error_url><ssl_test_mode>false</ssl_test_mode></txn>

Response forwarded to the receipt approval/declined URL

https://www.merchanturl.com/approved.html?xmldata=<txn><ssl_result>0</ssl_result><ssl_result_message>APPROVED</ssl_result_message><ssl_card_number>11********1111</ssl_card_number><ssl_exp_date>1208</ssl_exp_date><ssl_amount>2.34</ssl_amount>... (other xml fields)
  <ssl_txn_id>14039F9DA-1428-B5BB-508B-634707DC3413</ssl_txn_id><ssl_approval_code>N35032</ssl_approval_code><ssl_cvv2_response>P</ssl_cvv2_response><ssl_avs_response>X</ssl_avs_response><ssl_account_balance>0.00</ssl_account_balance><ssl_txn_time>10/26/2006 12:35:03 PM</ssl_txn_time></txn>

Example 4 – Electronic Gift Card

The following code example shows the differences in submitting an EGC transaction. The data below is a submission for a sale transaction. For Activation and Reload transactions an additional field will need to be submitted. <ssl_egc_tender_type>1</ ssl_egc_tender_type> (see appendix 1 for ssl_egc_tender_type values).

```
…/VirtualMerchant/processxml.do?xmldata=<txn><ssl_merchant_ID>123456</ssl_merchant_ID><ssl_us
er_id>123456</ssl_user_id><ssl_pin>V6NJ3A</ssl_pin><ssl_transaction_type>egcsale</ssl_transaction
_type><ssl_card_number>1111111111111111</ssl_card_number><ssl_exp_date>1210</ssl_exp_date>
<ssl_amount>2.34</ssl_amount><ssl_invoice_number>1234</ssl_invoice_number><ssl_first_name>cust
omer</ssl_first_name><ssl_last_name>name</ssl_last_name><ssl_avs_address>1234 main
st.</ssl_avs_address><ssl_address2>apt b</ssl_address2><ssl_city>any
town</ssl_city><ssl_state>ST</ssl_state><ssl_avs_zip>55555</ssl_avs_zip><ssl_phone>555-555-
5555</ssl_phone><ssl_email>customer@email.com</ssl_email><ssl_receipt_link_method>REDG</ssl_r
eceipt_link_method><ssl_receipt_link_url>https://www.merchanturl.com/receipt.html</ssl_receipt_link_url
><ssl_error_url>https://merchanturl.com/errorpage.html</ssl_error_url></txn>
```

Response forwarded to the receipt URL (formatted for easier viewing)

```
https://www.merchanturl.com/receipt.html?xmldata=- <txn>
 <ssl_card_number>50********3003</ssl_card_number>
 <ssl_exp_date>1208</ssl_exp_date>
 <ssl_amount>1.00</ssl_amount>
.
. other xml fields
.

 <ssl_result>0</ssl_result>
 <ssl_result_message>APPROVED</ssl_result_message>
 <ssl_txn_id>1DA26FC87-A4A1-65D7-83AD-B13C5F6AA8D5</ssl_txn_id>
 <ssl_approval_code>N50383</ssl_approval_code>
 <ssl_account_balance>10.00</ssl_account_balance>
 <ssl_txn_time>11/28/2006 12:50:40 PM</ssl_txn_time>
 </txn>
```

Example 5 – PinLess Debit

This data in this example is for a PinLess Debit transaction. This type of transaction is almost the same as examples 1-3 with a couple of fields needed in addition, ssl_customer_number and ssl_account_type. This authorizes the card as a debit transaction.

```
…/VirtualMerchant/processxml.do?xmldata=<txn><ssl_merchant_ID>111111</ssl_merchant_ID><ssl_pin
>111111</ssl_pin><ssl_transaction_type>ccsale</ssl_transaction_type><ssl_card_number>1111111111
111111</ssl_card_number><ssl_exp_date>1210</ssl_exp_date><ssl_amount>2.10</ssl_amount>
<ssl_customer_number>12345</ssl_customer_number><ssl_account_type>checking</ssl_account_type
> …</txn>
```

Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <txn>
  <ssl_card_number>11********1111</ssl_card_number>
  <ssl_amount>2.10</ssl_amount>
  <ssl_base_amount>2.10</ssl_base_amount>
  <ssl_surcharge_amount>0.00</ssl_surcharge_amount>
  <ssl_account_type>Checking</ssl_account_type>
  <ssl_customer_number>1111</ssl_customer_number>
.
.  other xml fields
.
  <ssl_result>0</ssl_result>
  <ssl_result_message>APPROVED</ssl_result_message>
  <ssl_txn_id>13745C6FE-C961-333B-763A-BE3A9E672321</ssl_txn_id>
  <ssl_approval_code>605630</ssl_approval_code>
  <ssl_account_balance>0.00</ssl_account_balance>
  <ssl_txn_time>11/28/2006 12:47:36 PM</ssl_txn_time>
  </txn>
```

Example 6 – Dynamic Currency Conversion

For merchants participating in the DCC program, the following example should be reviewed. Unlike DCC transactions that are submitted to process.do and receive a page for the decision, in XML format the merchant is responsible for providing the DCC decision page to the customer. The initial request will be the same as in examples 1-3, however a different response will be received, and the transaction is not complete at this point. Virtual Merchant requires a response before completing the transaction. You have 10 minutes to return a response. If the request is sent after 10 minutes, an error response will indicate that the transaction was not found and you will have to begin the transaction again. When the response is received within the 10-minute time frame the transaction completes and an authorization response message is provided as in examples 1-3. Any error, including unknown/missing ID or timeout messages, will be directed to the error URL, if one is specified. If an error URL is not specified the error will follow the normal response pattern (approval/decline response) for that particular Option. A separate URL for DCC decisions can be specified. The variable is called ssl_dcc_popup_url. There are 4 different ways to integrate this each listed as an option below.

Normal request sent to processxml.do

```
…/VirtualMerchant/processxml.do?xmldata=<txn><ssl_merchant_ID>111111</ssl_merchant_ID><ssl_pin>111111</ssl_pin><ssl_transaction_type>ccsale</ssl_transaction_type><ssl_card_number>1111111111111111</ssl_card_number><ssl_exp_date>1210</ssl_exp_date><ssl_amount>2.10</ssl_amount>…</txn>
```

processxml.do response:

OPTION A (receipt option LINK, GET, or POST, no ssl_dcc_popup_url defined)
Example
```
<?xml version="1.0" encoding="UTF-8" ?>
 - <txn>
    <id>5ZvaUB7w5kIzYJQfyASNkN3</id>
    <ssl_txn_currency_code>EUR</ssl_txn_currency_code>
    <ssl_conversion_rate>0.82</ssl_conversion_rate>
    <ssl_amount>1.00</ssl_amount>
    <ssl_cardholder_amount>0.82</ssl_cardholder_amount>
     - <dccoption>
         <option label="Please charge my purchase in my home currency">Y</option>
         <option label="Do not charge me in my home currency; charge my purchase in US
dollars">N</option>
      </dccoption>
  </txn>
```

Data sent to processxml.do
Example:
```
…/VirtualMerchant/processxml.do?xmldata=<txn><id>vEDlxmdYf4n3hN1IRLINynR</id><dccoption>Y</dccoption></txn>
```

processxml.do response.
Example:
```
<?xml version="1.0" encoding="UTF-8" ?>
- <txn>
 <ssl_result>0</ssl_result>
 <ssl_result_message>APPROVED</ssl_result_message>
```

```
  <ssl_card_number>11********1111</ssl_card_number>
  <ssl_exp_date>1208</ssl_exp_date>
  <ssl_amount>1.00</ssl_amount>
  <ssl_conversion_rate>0.82</ssl_conversion_rate>
  <ssl_cardholder_amount>0.82</ssl_cardholder_amount>
.
. (other xml fields, including any dcc response fields needed)
.
  <ssl_txn_id>14039F9DA-1428-B5BB-508B-634707DC3413</ssl_txn_id>
  <ssl_approval_code>N35032</ssl_approval_code>
  <ssl_cvv2_response />
  <ssl_avs_response />
  <ssl_account_balance>0.00</ssl_account_balance>
  <ssl_txn_time>10/26/2006 12:35:03 PM</ssl_txn_time>
  </txn>
```

END OPTION A

/VirtualMerchant/processxml.do response:

OPTION B (receipt option LINK, GET, or POST, ssl_dcc_popup_url defined)
A response data stream is returned to the ssl_dcc_popup_url as a GET request.
Example

```
http://www.merchanturl.com/dccpopupurl.html?xmldata=<txn><id>5ZvaUB7w5klzYJQfyASNkN3</id><ssl_txn_currency_code>EUR</ssl_txn_currency_code><ssl_conversion_rate>0.82</ssl_conversion_rate><ssl_amount>1.00</ssl_amount><ssl_cardholder_amount>0.82</ssl_cardholder_amount><dccoption><option label="Please charge my purchase in my home currency">Y</option><option label="Do not charge me in my home currency; charge my purchase in US dollars">N</option></dccoption></txn>
```


Sending Device: Data stream sent to processxml.do
Example

```
…/VirtualMerchant/processxml.do?xmldata=<txn><id>vEDlxmdYf4n3hN1IRLINynR</id><dccoption>Y</dccoption></txn>
```

processxml.do response
A response data stream is returned to the requesting device.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <txn>
  <ssl_result>0</ssl_result>
  <ssl_result_message>APPROVED</ssl_result_message>
  <ssl_card_number>50********3003</ssl_card_number>
  <ssl_exp_date>1208</ssl_exp_date>
  <ssl_amount>1.00</ssl_amount>
  <ssl_conversion_rate>0.82</ssl_conversion_rate>
  <ssl_cardholder_amount>0.82</ssl_cardholder_amount>
.
. (other xml fields, including any dcc response fields needed)
.
  <ssl_txn_id>14039F9DA-1428-B5BB-508B-634707DC3413</ssl_txn_id>
  <ssl_approval_code>N35032</ssl_approval_code>
  <ssl_cvv2_response />
  <ssl_avs_response />
```

```
<ssl_account_balance>0.00</ssl_account_balance>
<ssl_txn_time>10/26/2006 12:35:03 PM</ssl_txn_time>
</txn>
```

END OPTION B

processxml.do response:

OPTION C (receipt option REDG, No ssl_dcc_popup_url defined)
A response data stream is returned to the requesting device.
Example

```
<?xml version="1.0" encoding="UTF-8" ?>
- <txn>
    <id>5ZvaUB7w5klzYJQfyASNkN3</id>
    <ssl_txn_currency_code>EUR</ssl_txn_currency_code>
    <ssl_conversion_rate>0.82</ssl_conversion_rate>
    <ssl_amount>1.00</ssl_amount>
    <ssl_cardholder_amount>0.82</ssl_cardholder_amount>
    - <dccoption>
        <option label="Please charge my purchase in my home currency">Y</option>
        <option label="Do not charge me in my home currency; charge my purchase in US
dollars">N</option>
      </dccoption>
  </txn>
```

Sending Device: Data stream sent to processxml.do
Example

```
…/VirtualMerchant/processxml.do?xmldata=<txn><id>vEDlxmdYf4n3hN1IRLINynR</id><dccoption>Y</d
ccoption></txn>
```

processxml.do response
A response data stream is returned to the redirect url as a GET request.

Example

```
http://www.merchanturlhere.com/receivingpage.htm?xmldata=<txn><ssl_result>0</ssl_result><ssl_result
_message>APPROVED</ssl_result_message><ssl_card_number>11********1111</ssl_card_number><s
sl_exp_date>1208</ssl_exp_date><ssl_amount>1.00</ssl_amount><ssl_conversion_rate>0.82</ssl_con
version_rate><ssl_cardholder_amount>0.82</ssl_cardholder_amount>…  (other xml
fields)…<ssl_txn_id>14039F9DA-1428-B5BB-508B-
134707DC3413</ssl_txn_id><ssl_approval_code>N35032</ssl_approval_code><ssl_cvv2_response
/><ssl_avs_response /><ssl_account_balance>0.00</ssl_account_balance><ssl_txn_time>10/26/2006
12:35:03 PM</ssl_txn_time></txn>
```

END OPTION C

processxml.do response:

OPTION D (receipt option REDG, ssl_dcc_popup_url defined)
A response data stream is returned to the ssl_dcc_popup_url as a GET request.
Example

```
http://www.merchanturl.com/dccpopupurl.html?xmldata=<txn><id>5ZvaUB7w5klzYJQfyASNkN3</id><s
sl_txn_currency_code>EUR</ssl_txn_currency_code><ssl_conversion_rate>0.82</ssl_conversion_rate>
<ssl_amount>1.00</ssl_amount><ssl_cardholder_amount>0.82</ssl_cardholder_amount><dccoption><o
ption label="Please charge my purchase in my home currency">Y</option><option label="Do not
charge me in my home currency; charge my purchase in US dollars">N</option></dccoption></txn>
```

Sending Device: Data stream sent to processxml.do
Example

…/VirtualMerchant/processxml.do?xmldata=<txn><id>vEDlxmdYf4n3hN1IRLINynR</id><dccoption>Y</dccoption></txn>

processxml.do response
A response data stream is returned to the redirect url as a GET request.

Example

http://www.merchanturlhere.com/receivingpage.htm?xmldata=<txn><ssl_result>0</ssl_result><ssl_result_message>APPROVED</ssl_result_message><ssl_card_number>11********1111</ssl_card_number><ssl_exp_date>1208</ssl_exp_date><ssl_amount>1.00</ssl_amount><ssl_conversion_rate>0.82</ssl_conversion_rate><ssl_cardholder_amount>0.82</ssl_cardholder_amount>... (other xml fields)<ssl_txn_id>**14039F9DA-1428-B5BB-508B-134707DC3413**</ssl_txn_id><ssl_approval_code>**N35032**</ssl_approval_code><ssl_cvv2_response /><ssl_avs_response /><ssl_account_balance>**0.00**</ssl_account_balance><ssl_txn_time>**10/26/2006 12:35:03 PM**</ssl_txn_time></txn>

END OPTION D

Glossary of Terms

Address Verification

The process of verifying customer addresses with the issuing bank to minimize fraudulent transactions.

Authorization

The process of having credit card transactions approved by the issuing bank through communication with the NOVA Network.

Auto-Pend Transaction

A transaction option that automatically "Pends" Sale transactions submitted through the Virtual Merchant payment form.

Auto-Settle

An option that automatically settles all "unpended" transactions in the Unsettled Transaction batch at a specified time each day.

Card Verification Value

The process of verifying the Card Verification Value with the issuing bank to minimize fraudulent transactions. The CVV2 value is a three of four digit value that is printed in reverse italics on the back side of the card. This additional value is not embossed upon the front of the card, nor is it contained upon the magnetic stripe on back.

Comma-Separated Value

A text file format in which all data elements within the file are separated by a comma. This format is also referred to as a comma delimited file.

Filter

A function that allows you to enter specific parameters to narrow a search for transaction information in a particular file. You can search for a specific card number, within a specific date range, etc.

Force Transaction

A previously authorized transaction that needs to be entered in the current batch.

GBOK Number

A successful settlement batch with the NOVA Network.

Merchant Admin

The default user account for the Virtual Merchant account; the **Merchant Admin User ID (MA)** is the same as the **Virtual Merchant Account ID**. This special user cannot be deleted, always has all user rights and all terminal associations.

Peer User

A user who shares the same supervisor as you.

Pend Transaction

A transaction status option that will not allow the transaction to be submitted for settlement. To change the status of the transaction to allow it to be submitted for settlement, it must be "**Unpended**".

Refund Transaction

A transaction used to refund a previous purchase.

Sale Transaction

A transaction in which an authorization is obtained and the transaction is entered into the unsettled batch.

Scope of user rights

Virtual Terminal and Terminal Setup rights apply to your ability to do things in the context of any terminal in your Terminal Associations list. User Management rights apply to your ability to do things to your subordinates and to your peers' subordinates. If you have the Edit Terminal Associations right, you may only add terminal associations that you, yourself have.

Settlement Process

The process of sending a batch of previously authorized transactions for settlement to the NOVA Network.

Subordinate

This is anyone who is directly below you in the user hierarchy, or any of their **subordinates**.

Supervisor

This is the person directly above you in the user hierarchy.

Tab-Delimited Value

A text file format in which all data elements within the file are separated by the Tab character.

Terminal Association

Where your user rights refer to something you can do involving a terminal (make a sale or settle a transaction), your user must be associated with that terminal and you must have selected that terminal context in Virtual Merchant. See the chapter on User Management for details on how to make or edit **Terminal Associations**.

Terminal Friendly Name

Terminals are referred to in Virtual Merchant by a **Friendly Name** configured by NOVA's Internet Product Support, for instance, "Website Terminal."

Terminal ID

A number used to identify the source of a transaction to the NOVA Network. This corresponds to a physical credit card terminal in a traditional POS solution, but for Virtual Merchant, this is a virtual ID. You may have more than one terminal for use within your Virtual Merchant account. Each **Terminal ID (TID)** is associated with certain features as dictated by your NOVA merchant agreement. "Merchant Information" in Terminal Setup can be different for each terminal so that, for instance, the address printed on a receipt is correct for that location. See the chapter on Terminal Setup for details on configuring your terminal.

Unpend Transaction

A transaction status option that allows the transaction to be submitted for settlement. To prohibit the transaction from being submitted for settlement, it must be "**Pended**".

User Account

The **use**r you use to sign in to Virtual Merchant; the **User ID** is case sensitive

User Rights

The tasks that your **User Account** can do in Virtual Merchant; there are three areas of **User Rights**: Virtual Terminal, User Management and Terminal Setup. See the section on User Management for details on how to make or edit **User Rights**

Virtual Merchant Account

The **Virtual Merchant Account** your company has with the NOVA Network.

Developer Guide
3/16/2007

NOVA NETWORK
The most reliable payment processing
network in the industry