Authorize.Net®
a CyberSource solution

Server Integration Method (SIM)

Developer Guide

Card Not Present Transactions

Authorize.Net Developer Support

Authorize.Net Trademarks

Advanced Fraud Detection Suite™

Authorize.Net®

Authorize.Net Your Gateway to IP Transactions™

Authorize.Net Verified Merchant Seal™
Authorize.Net Where the World Transacts®

Automated Recurring Billing™

eCheck.Net®

FraudScreen.Net®

Last revised: 8/30/2010

Table of Contents

Last revised: 8/30/2010

SIM Developer Guide

Last revised: 8/30/2010

Revision History

PUBLISH DATE	UPDATES
August 2007	Release of Ver 2.0 Server Integration Method (SIM) Developer Guide
May 2008	Remove SecureSource requirements and various updates
July 2009	Clarify use of x_recurring_billing, x_fp_timestamp, x_version
	Added warning for Merchant Defined Fields
	Additions to the Response Code list
	Updates to Silent Post URL
	New section Renaming a Field
September 2009	Added note explaining new policy with respect to authorization transactions for Visa credit cards
	Corrected list of supported graphics formats for the Hosted Payment form. (Removed SWF as a supported format)
	Deprecation of transaction types Prior_auth_Capture, Voids, Credits
October 2009	Update definition of x_type to reflect deprecation of transaction types Prior_auth_capture, Credit, and Void.
	Correct capitalization of the *eCheck.Net Developer Guide*.
December 2009	Updated reason-response codes
	Clarified transaction types
February 2010	Updated procedure for uploading images for hosted page
May 2010	Correct references to transaction ID, and non-supported transaction types (Credit, Void, and Prior Auth Capture
August 2010	Added information about configuring look and feel of payment form and receipt page.

Section 1
Introduction

Welcome to the Authorize.Net *Server Integration Method (SIM) Developer Guide*. This guide describes the Web development required to connect an e-commerce website or other application to the Authorize.Net Payment Gateway in order to submit credit card transactions for authorization and settlement using SIM.

SIM is a hosted payment processing solution that handles all the steps in processing a transaction, including:

- Collection of customer payment information through a secure, hosted form
- Generation of a receipt to the customer
- Secure transmission to the payment processsing networks for settlement
- Funding of proceeds to the merchant's bank account
- Secure storage of cardholder information

The security of a SIM transaction is assured through the use of a unique digital signature or "fingerprint" that is sent with each transaction. This fingerprint is used by Authorize.Net to authenticate both the merchant and the transaction. Sample code for this function is available for free from the Authorize.Net Developer Center at http://developer.authorize.net.

SIM is an ideal integration solution because merchants are not required to collect, transmit or store sensitive cardholder information to process transactions. Additionally, SIM does not require merchants to purchase and install a Secure Sockets Layer (SSL) digital certificate. This removes the complexity of securely handling and storing cardholder information, simplifying compliance with the Payment Card Industry (PCI) Data Security Standard.

Note: For merchants who need a highly customizable payment form (for example, complete control of look and feel and the ability to keep the customer on their website during the entire checkout process) or who are integrating a standalone business application, Authorize.Net recommends the Advanced Integration Method (AIM). The *AIM Developer Guide* can be found at http://developer.authorize.net/guides/AIM/.

SIM Minimum Requirements

Before you begin, check with the merchant to make sure that the following SIM requirements have already been met. It is strongly recommended that you work closely with the merchant to ensure that any other business and website requirements (for example, bank or processor requirements, website design preferences) are included in their SIM integration.

- The merchant must have a U.S.-based merchant bank account that allows Internet transactions.
- The merchant must have an e-commerce (Card Not Present) Authorize.Net Payment Gateway account.
- The merchant's website must be capable of performing an HTML Form POST to request the secure payment gateway hosted payment form.
- The merchant's website or hosting provider must have server scripting or CGI capabilities such as ASP Classic, Cold Fusion, PHP or Perl.
- The merchant must be able to store payment gateway account data securely (for example, API Login ID or Transaction Key).

Note: Merchants should avoid storing any type of sensitive cardholder information. However, in the event that a merchant or third party must store sensitive customer business or payment information, compliance with industry standard storage requirements is required. Please see the *Developer Security Best Practices White Paper* at http://www.authorize.net/files/developerbestpractices.pdf for guidelines

Managing Integration Settings

When integrating to the payment gateway, you should be aware that most settings for a merchant's integration can be configured and managed in two ways:

- Included in the transaction request on a per-transaction basis by means of the application programming interface (API), (as described in this guide), OR
- Configured in the Merchant Interface and applied to all transactions.

IMPORTANT: The Merchant Interface at https://secure.authorize.net is a secure website where merchants can manage their payment gateway account settings, including their website integration settings. It is recommended that you review the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for information on managing the merchant's payment gateway integration using the Merchant Interface.

Transaction settings submitted in the transaction request override transaction settings configured in the Merchant Interface. However, please be aware that some integration settings must be configured in the Merchant Interface. To help the merchant maintain a robust integration, you should review the integration settings that can be configured in the Merchant Interface with the merchant and determine which integration settings can be posted on a per-transaction basis, and which should be configured in the Merchant Interface. See "Appendix A Fields by Transaction Type" for a list of fields the payment gateway recommends be submitted on a per-transaction basis.

Features of SIM

In addition to basic transaction processing, SIM provides merchants with several features for configuring transaction security options and further customizing their customers' checkout experience. These features are listed in the SIM Feature Selection Guide provided below. Please take a few moments to discuss these with your merchant and select which features they would like to include in their integration.

FEATURE	DESCRIPTION	REQUIREMENTS
Address Verification Service (AVS) Filter	This feature allows merchants to compare the billing address submitted by the customer for the transaction with the address on file at the card issuing bank. Filter settings in the Merchant Interface allow the merchant to reject transactions based on the AVS response received.	To implement AVS, the merchant must require the Address and ZIP Code fields on the payment gateway hosted payment form. For more information about AVS, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant.
Card Code Verification (CCV) Filter	This feature allows merchants to compare the card code submitted by the customer for the transaction with the card code on file at the card issuing bank. Filter settings in the Merchant Interface allow the merchant to reject transactions based on the CCV response received.	To implement CCV, the merchant must require the Card Code field on the payment gateway hosted payment form. For more information CCV, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant.
Itemized Order Information	This feature allows merchants to submit details for items purchased. This information is included in the merchant transaction confirmation email, in the Transaction Details for the transaction, and in QuickBooks download reports in the Merchant Interface.	To implement Itemized Order Information, the line item field must be submitted on a per-transaction basis. Please see the "Itemized Order Information" section of this document for details.
Receipt Page	This feature allows merchants to customize the payment gateway hosted receipt page that is displayed to the customer at the completion of a transaction. This page can include a hyperlink back to the merchant's website.	To configure the payment gateway hosted receipt page, settings must be configured in the Receipt Page section of the Settings menu in the Merchant Interface or submitted on a per-transaction basis. Please see the "Receipt Options" section of this document for details.

Last revised: 8/30/2010

Email Receipt	This feature allows merchants to select an automatic email receipt to be sent by the payment gateway to their customers.	To configure the payment gateway email receipt, the merchant must require the customer email address on the hosted payment form, and settings must be configured in the Email Receipts section of the Settings menu in the Merchant Interface or submitted on a per-transaction basis. Please see the "Receipt Options" section of this document for details.
Relay Response	This feature allows merchants to choose to display a more customized receipt page that is generated on the merchant's Web server and relayed by the payment gateway to the customer's browser.	To configure Relay Response, settings must be configured in the Relay Response section of the Settings menu in the Merchant Interface or submitted on a per-transaction basis. Please see the "Relay Response" section of this document for details.

eCheck.Net®

In addition to processing credit card transactions, the payment gateway also supports electronic check transactions with our exclusive eCheck.Net® product. Please contact the merchant to determine whether eCheck.Net is enabled for their payment gateway account or if they would like to sign up. **In the event that eCheck.Net is enabled, you will need to ensure that the merchant's website integration supports all eCheck.Net field requirements.** Please see the *eCheck.Net Developer Guide* at http://developer.authorize.net/guides/echeck.pdf for more information.

Developer Support

There are several resources available to help you successfully integrate a merchant website or other application to the Authorize.Net Payment Gateway.

The Developer Center at http://developer.authorize.net provides test accounts, sample code, FAQs, and troubleshooting tools.

If you can't find what you need in the Developer Center, our Integration Team is available to answer your questions by email at integration@authorize.net. (Our Integration Team can only assist with support requests specifically about the Authorize.Net application programming interface (API) and/or services.)

Be sure to read our *Developer Security Best Practices White Paper* at http://www.authorize.net/files/developerbestpractices.pdf for information on how to maximize the security and reliability of your merchant integration solutions.

If you have any suggestions about how we can improve or correct this guide, please email documentation@authorize.net.

Last revised: 8/30/2010

Section 2
Transaction Data Requirements

The payment gateway supports several credit card transaction types for transactions submitted by means of SIM.

To implement SIM for a merchant's website, you need to develop an HTML Form POST to request the secure payment gateway hosted payment form and pass required and optional merchant and transaction information.

To see a table listing minimum form field requirements for posting credit card transaction requests to the payment gateway, see Requesting the Secure Hosted Payment Form on page 15.

Credit Card Transaction Types

This section describes the credit card transaction types supported by the payment gateway and their specific field requirements. It's a good idea to talk to your merchant about how their business plans to submit transaction so that you can properly integrate their payment gateway account to support their business processes.

For example, are they submitting transactions mainly through an e-commerce website? Do they need to integrate a custom application to allow call center representatives to enter mail order/telephone order (MOTO) transactions? Would they like the ability to verify the availability of funds on a customer's credit card account at the time of purchase and then charge the credit card at the time they ship the order?

The payment gateway supports the following credit card transaction types.

Note:	Some of the field requirements listed in this section for each credit card transaction type are *in addition* to the minimum field requirements already set forth above for ALL transactions submitted to the payment gateway. For a list of all fields that are required for each credit card transaction type, please see Appendix A: Fields by Transaction Type" in this document.

Authorization and Capture

Authorization with Auto Capture (Auth_Capture) is the default transaction type in the Virtual Terminal. If no *x_type* variable is submitted with a website transaction request, the type defaults to `Auth_Capture`. This type of transaction is completely automatic: the transaction is submitted to your processor for authorization and, if approved, is placed in your Unsettled Transactions already set to Capture. The transaction will settle out with your next batch settlement (settlement occurs every 24 hours, within 24 hours of the time specified in your Settings menu, under **Transaction Cutoff Time**).

The unique field requirement for an Authorization and Capture is:

```
<INPUT TYPE=HIDDEN NAME="x_type" VALUE="AUTH_CAPTURE">
```

Authorization Only

This transaction type is sent for authorization only. When an Authorization Only (Auth_Only) transaction is submitted, it is sent to your processor for authorization. If approved, the transaction is placed in your Unsettled Transactions with a status of Authorized/Pending Capture. The authorization places the funds on hold with the customer's bank, but until the transaction is captured, the funds transfer process does not take place. This type of transaction is not sent for settlement until you submit a credit card transaction type Prior Authorization and Capture, or if you submit the transaction for capture manually in the Merchant Interface. This can be useful in situations where you need to make a sale, but won't be able to ship merchandise for several days; you can authorize the transaction to ensure the availability of funds, then, once you have shipped, you can capture the transaction to obtain the funds.

Authorization Only transactions are only kept in your Unsettled Transactions for 30 days. After that, its transaction status changes to Expired, and the funds will NOT be transferred. To capture a transaction, you can manually log on to your Authorize.Net interface and go to your Unsettled Transactions. From there, you can use the Group Capture filter to capture multiple transactions at once, or click on the individual Transaction ID of the transaction you would like to capture, and the next screen will provide a Capture button. From a website or billing application, you can submit the x_Type variable, with a value of Prior_Auth_Capture, to capture the transaction.

The unique field requirement for an Authorization Only is:

```
<INPUT TYPE=HIDDEN NAME="x_type" VALUE="AUTH_ONLY">
```

Note: Merchants who use SIM can configure the hosted payment form to submit either Authorization and Capture or Authorization Only transactions. Please check with the merchant regarding their preferences regarding which of these credit card transaction types should be used for their website.

Prior Authorization and Capture

This transaction type is used to complete an Authorization Only transaction that was successfully authorized through the payment gateway. It can only be submitted from the Merchant Interface, not from a SIM application.

If this transaction type is required, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Capture Only

Capture_Only transactions are used when you already have an authorization from a bank. To use this type of transaction, you need an authorization code from the card issuer (usually a five- or six-digit number). For example, if you called Visa directly and obtained an authorization over the phone, you would need to submit a Capture_Only transaction to start the funds transfer process. You can manually submit a Capture_Only transaction from your Virtual Terminal of the Merchant Interface by selecting Capture Only, or from a website or billing application by including the following variables with your transaction request:

- `x_Type` (Capture_Only)
- `x_Auth_Code` (the five- or six-digit code provided by the card issuer)

Credit

This transaction type is used to refund a customer for a transaction that was originally processed and successfully settled through the payment gateway. The SIM API does not support Credit transactions. Credits can be manually processed through the Virtual Terminal of the Merchant Interface.

Last revised: 8/30/2010

If this transaction type is required, the merchant should process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Void

This transaction type is used to cancel an existing transaction that has a status of Authorized/Pending Capture or Captured/Pending Settlement. Settled transactions cannot be voided (issue a Credit to reverse such charges). The SIM API does not support Void transactions.

You can manually void transactions from the Unsettled Transactions screen of the Merchant Interface. From there, you can use the Group Void filter toward to the top of your screen to void multiple transactions at once, or click on the individual Transaction ID of the transaction you would like to void, and the next screen will provide a Void button. They can also be submitted from a website or billing application that uses AIM.

If this transaction type is required, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Using the Merchant Interface

The Merchant Interface allows merchants to manage transactions, capture Authorize Only transactions, void transactions, and issue refunds. These transaction types can also be managed automatically by means of the API, if you are integrating a custom application to the payment gateway. However, for most integrations, these transaction types can be more conveniently and easily managed in the Merchant Interface.

For more information on submitting transactions in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant or click **Help** in the top right corner of the Merchant Interface.

Last revised: 8/30/2010

Section 3
Submitting Transactions

The standard payment gateway application programming interface (API) consists of required and additional optional form fields that can be submitted to the payment gateway for real-time transaction processing. The API includes fields for requesting the payment gateway's secure hosted payment form, which can be customized to reflect the look and feel of the merchant's website.

Transaction Post Location

The merchant's website should post transaction requests by means of an HTML Form POST to the following payment gateway URL:

```
https://secure.authorize.net/gateway/transact.dll
```

Generating the Unique Transaction Fingerprint

The transaction authentication piece for the Server Integration Method is a "transaction fingerprint," or a hash of merchant- and transaction-specific information using the HMAC-MD5 (MD5 RFC 1321 with a 128-bit hash value) hashing algorithm. The HMAC-MD5 is used only for generating the unique transaction fingerprint.. The transaction fingerprint must be generated on a per-transaction basis by a server-side script on the merchant's Web server, and inserted into the transaction request. The payment gateway uses the same mutually exclusive merchant information to decrypt the transaction fingerprint and authenticate the transaction.

There are two options for developing the fingerprint generation script:

- You can develop a custom script yourself using the API fields information in this section, OR

- You can use Authorize.Net sample code available for free from our Developer Center at http://developer.authorize.net.

Custom transaction fingerprint code

If you choose to develop custom code for generating the transaction fingerprint, the following table represents the field requirements for the transaction fingerprint. The form fields inserted into the transaction request by the fingerprint generation use the following syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the field">
```

FIELD NAME	VALUE	FORMAT	NOTES

FIELD NAME	VALUE	FORMAT	NOTES
x_fp_hash	The transaction unique fingerprint	N/A	The fingerprint is generated using the HMAC-MD5 hashing algorithm on the following field values: • API Login ID (x_login) • The sequence number of the transaction (x_fp_sequence) • The timestamp of the sequence number creation (x_fp_timestamp) • Amount (x_amount) Field values are concatenated and separated by the "^" character.
x_fp_sequence	The merchant assigned sequence number for the transaction	Numeric	The sequence number can be a merchant assigned value, such as an invoice number or any randomly generated number.
x_fp_timestamp	The timestamp at the time of fingerprint generation.	UTC time in seconds since January 1, 1970	Coordinated Universal Time (UTC) is an international atomic standard of time (sometimes referred to as GMT). Using a local time zone timestamp will cause fingerprint authentication to fail. If the fingerprint is more than one hour old or more than 15 minutes into the future, it is rejected.

The transaction fingerprint that is submitted in *x_fp_hash* is generated using an HMAC-MD5 hashing algorithm on the following field values:

1. API Login ID (*x_login*)
2. Sequence number (*x_fp_sequence*)
3. UTC timestamp in seconds (*x_fp_timestamp*)

> **Note:** **Be sure that the merchant server's system clock is set to the proper time and time zone.**

4. Amount (*x_amount*)

> **Note:** **The amount used to generate the fingerprint must reflect the final amount of the transaction. To avoid any discrepancy, it is strongly recommended that you generate the fingerprint at a point in the checkout process when the amount can no longer be changed.**

When generating the fingerprint, input values must be provided to the script in the field order listed above and concatenated by the "^" character. All trailing spaces must be removed from input

Last revised: 8/30/2010

values. If the fingerprint is generated using any other field order, authentication will fail and the transaction will be rejected.

Sample 1. Example of fingerprint input field order

```
"authnettest^789^67897654^10.50^"
```

Please note the required trailing "^" character

The Transaction Key

The cryptographic key used in the HMAC calculation is the merchant's unique Transaction Key, which is a payment gateway-generated, 16-character value that can be obtained by the merchant in the Merchant Interface. For more information, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant.

IMPORTANT: The merchant's Transaction Key is highly sensitive and should only be known by the payment gateway and the merchant. For this reason it is vital that the Transaction Key is stored securely and separately from the merchant's Web server. In addition, please note that the merchant's API Login ID will be visible in the source for the payment form request, but the Transaction Key should never be visible.

Sample 2. Example of the call to generate the transaction fingerprint

```
Fingerprint = HMAC-MD5
("authnettest^789^67897654^10.50^","abcdefgh12345678")
```

Requesting the Secure Hosted Payment Form

To display the payment gateway hosted payment form to a customer, a payment form request needs to be submitted using an HTML Form POST with hidden fields. The following table describes the minimum fields required for requesting the hosted payment form. The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the
field">
```

FIELD NAME	VALUE	FORMAT	NOTES
x_login	The merchant's unique API Login ID	Up to 20 characters	The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and transaction fingerprint together provide the merchant authentication required for access to the payment gateway. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant for more information.
x_type	The type of credit card transaction	AUTH_CAPTURE (default), AUTH_ONLY	If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an AUTH_CAPTURE.
x_amount	The amount of the transaction	Up to 15 digits with a decimal point (no dollar symbol) Ex. 8.95	This is the total amount and must *include* tax, shipping, and any other charges. The amount can be either hard coded or posted to a script.
x_show_form	The payment form request	PAYMENT_FORM	The show form field indicates that the merchant would like to use the payment gateway hosted payment form to collect payment data.
x_trans_id	The payment gateway assigned transaction ID of an original transaction		Required only for CREDIT, PRIOR_AUTH_CAPTURE, and VOID transactions submitted using AIM. For more information about transaction types, see the "Credit Card Transaction Types" section of this document.
x_relay_response	Indicates whether a relay response is desired	TRUE	This field instructs the payment gateway to return transaction results to the merchant by means of an HTML form POST to the merchant's Web server for a relay response.

Last revised: 8/30/2010

Section 4 Receipt Options

FIELD NAME	VALUE	FORMAT	NOTES
x_delim_data	FALSE	FALSE, F, NO, N, 0	This field is used for AIM. Set it to FALSE if you are using SIM.
x_fp_hash	The transaction unique fingerprint	N/A	The fingerprint is generated using the HMAC-MD5 hashing algorithm on the following field values:

API Login ID (x_login)

The sequence number of the transaction (x_fp_sequence)

The timestamp of the sequence number creation (x_fp_timestamp)

Amount (x_amount)

Field values are concatenated and separated by the "^" character. |
x_fp_sequence	The merchant assigned sequence number for the transaction	Numeric	The sequence number can be a merchant assigned value, such as an invoice number or any randomly generated number.
x_fp_timestamp	The timestamp at the time of fingerprint generation.	UTC time in seconds since January 1, 1970	Coordinated Universal Time (UTC) is an international atomic standard of time (sometimes referred to as GMT). Using a local time zone timestamp will cause fingerprint authentication to fail.
x_show_form	The payment form request	PAYMENT_FORM	The show form field indicates that the merchant would like to use the payment gateway hosted payment form to collect payment data.

The sample code below is an example of the minimum requirements for requesting the hosted payment form and produces a button (Figure 1.) that is displayed to the customer upon checkout. When the customer clicks the button, the secure hosted payment form (Figure 2.) is displayed in the customer's browser.

The code also shows that the fingerprint hash function inserts the required input fields into the HTML Form POST. Ideally, once the button is clicked, the following should occur:

1. The sequence number is generated.
2. The final total amount of the transaction is calculated.
3. A call is made to server-side script on the merchant's Web server that generates the transaction fingerprint (InsertFP).
4. The payment gateway hosted payment form request is sent to the payment gateway.

Note: The sample code included in this document uses dummy field values in an ASP scripting environment. Because code varies based on the Web programming language used, it is not recommended that you copy and paste sample code but rather use it as a guide. Additional sample code is available for download from the Authorize.Net Developer Center at http://developer.authorize.net.

Sample 3. Submitting a request for the hosted payment form request

```
<!--#INCLUDE FILE="simlib.asp"-->

<FORM METHOD=POST ACTION=
"https://secure.authorize.net/gateway/transact.dll">
    <% ret = InsertFP (APIloginid, sequence, amount, txnkey)
%>
<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">

<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the merchant's API
Login ID">

<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">

<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">

<INPUT TYPE=SUBMIT VALUE="Click here for the secure payment
form">
</FORM>
```

Even though the *x_version* field included in the sample above is not technically a minimum requirement for submitting a transaction, it is highly recommended that you submit this field on a per-transaction basis, particularly if you are using Relay Response. For more information, see the "Additional API Fields" and "Best Practices Fields" sections of this document.

Note: The use of frames with the hosted payment form is not recommended. Even though the hosted payment form is secure, the lock icon on the user's status bar will key off the surrounding frame and not the payment form. The page will not look secure to the customer.

Figure 1. The payment form button

Click here for the secure payment form

Last revised: 8/30/2010
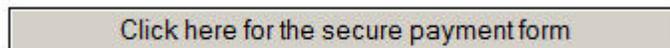
Figure 2. The secure hosted payment form

By default, the hosted payment form always displays the fields required to post a credit card transaction:

- Amount
- Credit Card Number
- Expiration Date

Customizing the hosted payment form fields

Although the sample HTML code included in the previous section is sufficient to request the payment gateway hosted payment form, additional fields can be configured for the payment form in the Merchant Interface, or submitted with the HTML Form POST. This allows the merchant to display a more detailed payment form and/or collect additional information from the customer.

IMPORTANT: Regardless of how additional fields are configured for the payment form, the following attributes **must** also be configured for additional fields in the Merchant Interface in order to be displayed properly on the hosted payment form.

- **View** – The customer can view but not edit the information. For example, if the merchant would like to display an invoice number.
- **Edit** – The customer can view and edit the information but the field is not required for the transaction. For example, if the merchant would like to collect but does not require the customer's email address.
- **Required** – The customer is required to provide information in the field to submit the transaction. For example, if the merchant would like to require the customer's card code.

Note: These field attributes only dictate what is displayed on the hosted payment form. Any fields that are submitted with the HTML Form POST but do not have attributes configured in the Merchant Interface are still submitted with the transaction to the payment gateway. This

Last revised: 8/30/2010

allows the merchants requesting a Relay Response to receive back transaction or order information that is not necessary for the customer to view or submit. For more information on Relay Response, see the "Receipt Options" section of this guide.

For information on configuring payment form fields and attributes in the Merchant Interface, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant.

The following table lists the payment form fields that can be configured in the Merchant Interface or submitted by means of the payment form request. The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the
field">
```

FIELD NAME	VALUE	FORMAT	NOTES
PAYMENT INFORMATION			
Recurring Billing Transaction (x_recurring_billing)	The recurring billing status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicating marker used by merchant account providers to identify transactions which originate from merchant hosted recurring billing applications. This value is not affiliated with Automated Recurring Billing.
ORDER INFORMATION			
Invoice Number (x_invoice_num)	The merchant assigned invoice number for the transaction	Up to 20 characters (no symbols)	The invoice number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be included on the hosted payment form, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
Description (x_description)	The transaction description	Up to 255 characters (no symbols)	The description must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
BILLING INFORMATION			
First Name (x_first_name)	The first name associated with the customer's billing address	Up to 50 characters (no symbols)	
Last Name (x_last_name)	The last name associated with the customer's billing address	Up to 50 characters (no symbols)	
Company (x_company)	The company associated with the customer's billing address	Up to 50 characters (no symbols)	
Address (x_address)	The customer's billing address	Up to 60 characters (no symbols)	Required if the merchant would like to use the Address Verification Service filter. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant Required for Zero Dollar Authorizations for Visa verification transactions.
City (x_city)	The city of the customer's billing address	Up to 40 characters (no symbols)	
State (x_state)	The state of the customer's billing address	Up to 40 characters (no symbols) or a valid two-character state code	
ZIP Code (x_zip)	The ZIP code of the customer's billing address	Up to 20 characters (no symbols)	Required if the merchant would like to use the Address Verification Service filter. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant Required for Zero Dollar Authorizations for Visa verification transactions.
Country	The country of the	Up to 60	

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
(x_country)	customer's billing address	characters (no symbols)	
Phone (x_phone)	The phone number associated with the customer's billing address	Up to 25 digits (no letters) Ex. (123)123-1234	
FAX (x_fax)	The fax number associated with the customer's billing address	Up to 25 digits (no letters) Ex. (123)123-1234	
Email (x_email)	The customer's valid email address	Up to 255 characters Ex. janedoe@customer.com	The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid. For more information about Email Receipts, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/
Customer ID (x_cust_id)	The merchant assigned customer ID	Up to 20 characters (no symbols)	The unique identifier to represent the customer associated with the transaction. The customer ID must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
SHIPPING INFORMATION			
First Name (x_ship_to_first_na	The first name associated with the	Up to 50 characters (no	

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
me)	customer's shipping address	symbols)	
Last Name (x_ship_to_last_name)	The last name associated with the customer's shipping address	Up to 50 characters (no symbols)	
Company (x_ship_to_company)	The company associated with the customer's shipping address	Up to 50 characters (no symbols)	
Address (x_ship_to_address)	The customer's shipping address	Up to 60 characters (no symbols)	
City (x_ship_to_city)	The city of the customer's shipping address	Up to 40 characters (no symbols)	
State (x_ship_to_state)	The state of the customer's shipping address	Up to 40 characters (no symbols) or a valid two-character state code	
ZIP Code (x_ship_to_zip)	The ZIP code of the customer's shipping address	Up to 20 characters (no symbols)	
Country (x_ship_to_country)	The country of the customer's shipping address	Up to 60 characters (no symbols)	

ADDITIONAL SHIPPING INFORMATION (Level 2 Data)

FIELD NAME	VALUE	FORMAT	NOTES
Tax (x_tax)	The valid tax amount OR delimited tax information	When submitting delimited tax information, values must be delimited by a bracketed pipe <|>	The tax amount charged; or, when submitting this information by means of the HTML Form POST, delimited tax information including the sales tax name, description, and amount is also allowed. The total amount of the transaction in x_amount must *include* this amount.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
	tax item name<|>		The tax item name.
	tax description<|>		The tax item description.
	tax amount	The dollar sign ($) is not allowed when submitting delimited information.	The tax item amount. The total amount of the transaction in x_amount must *include* this amount.
Example:	`<INPUT TYPE="HIDDEN" name="x_tax" VALUE="Tax1<|>state tax<|>0.0625">`		
Freight (x_freight)	The valid freight amount, or delimited freight information	When submitting delimited freight information, values must be delimited by a bracketed pipe <|>	The freight amount charged; or, when submitting this information by means of the HTML Form POST, delimited freight information including the freight name, description, and amount is also allowed. The total amount of the transaction in x_amount must *include* this amount.
	freight item name<|>		The freight item name.
	freight description<|>		The freight item description.
	freight amount	The dollar sign ($) is not allowed when submitting delimited information.	The freight amount. The total amount of the transaction in x_amount must *include* this amount.
Example:	`<INPUT TYPE="HIDDEN" name="x_freight" VALUE="Freight1<|>ground overnight<|>12.95">`		
Duty (x_duty)	The valid duty amount OR delimited duty information	When submitting delimited duty information, values must be delimited by a bracketed pipe <|>	The duty amount charged; or, when submitting this information by means of the HTML Form POST, delimited duty information including the duty name, description, and amount is also allowed. The total amount of the transaction in x_amount must *include* this

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
			amount.
	duty item name<|>		The duty item name.
	duty description<|>		The duty item description.
	duty amount	The dollar sign ($) is not allowed when submitting delimited information.	The duty amount. The total amount of the transaction in x_amount must *include* this amount.
Example:	`<INPUT TYPE="HIDDEN" name="x_duty" VALUE="Duty1<|>export<|> 15.00>`		
Tax Exempt (x_tax_exempt)	The tax exempt status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates whether the transaction is tax exempt.
Purchase Order Number (x_po_num)	The merchant assigned purchase order number	Up to 25 characters (no symbols)	The purchase order number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.

IMPORTANT: If the merchant chooses to use the standard payment gateway security features, Address Verification Service (AVS) and Card Code Verification (CCV), they need to require the customer's card code and billing address information on the payment gateway hosted payment form. These requirements must be configured in the Payment Form setting in the Merchant Interface. For more information about AVS and CCV, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Note: Delimited duty, freight, and tax information are not returned in the transaction response or in the merchant confirmation email. This information is displayed only on the Transaction Detail page in the Merchant Interface.

The following HTML sample code is an example of requesting additional supported fields using the hosted payment form. In this sample, the payment form will display the Invoice Number, Description, Customer ID, billing information, and shipping information fields. These fields can also be configured for the payment form in the Merchant Interface.

Note:	The Invoice Number and Customer ID must be created dynamically or provided on a per-transaction basis in order to include this information in the post. The payment gateway does not perform this function.

For the purposes of this example, the Invoice Number, Description, and Customer ID fields have been previously configured in the Merchant Interface as **View**, and billing and shipping information fields have been configured as **Edit**.

Note:	The sample code included in this document uses dummy field values in an ASP scripting environment. Because code varies based on the Web programming language used, it is not recommended that you copy and paste sample code but rather use it as a guide. Additional sample code is available for download from the Authorize.Net Developer Center at http://developer.authorize.net.

Sample 4. Payment form request with additional transaction data

```
<!--#INCLUDE FILE="simlib.asp"-->
<FORM METHOD=POST ACTION=
"https://secure.authorize.net/gateway/transact.dll">
    <% ret = InsertFP (APIloginid, sequence, amount, txnkey) %>
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the merchant's API Login
ID">
<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">
<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">
<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_invoice_num" VALUE="ORDER-002450">
<INPUT TYPE=HIDDEN NAME="x_description" VALUE="Product or order
description.">
<INPUT TYPE=HIDDEN NAME="x_cust_id" VALUE="Doe-John 001">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">
<INPUT TYPE=SUBMIT VALUE="Click here for the secure payment
form">
</FORM>
```

The sample code above produces a button that is displayed to the customer upon checkout. When the customer clicks the button, the secure hosted payment form is displayed.

Last revised: 8/30/2010

Figure 3. The hosted payment form with additional form fields

Order Information   * Required Fields

Invoice Number: ORDER-002450
Description: Product or order description.
Total: US $9.95

Payment Information

Card Number: [   ] * (enter number without spaces or dashes)
Expiration Date: [   ] * (mmyy)

Billing Information

Customer ID: Doe-John 001
First Name: [   ]   Last Name: [   ]
Company: [   ]
Address: [   ]
City: [   ]
State/Province: [   ]   Zip/Postal Code: [   ]
Country: [   ]
Email: [   ]
Phone: [   ]
Fax: [   ]

Shipping Information

☐ Copy Billing Information to Shipping Information
First Name: [   ]   Last Name: [   ]
Company: [   ]
Address: [   ]
City: [   ]
State/Province: [   ]   Zip/Postal Code: [   ]
Country: [   ]

[Submit]   [Reset Form]

Customizing the hosted payment form look and feel

When using the hosted payment form, the following settings can be configured to match the look of the merchant's website:

- The color of the text
- The color of link text
- The background color
- The header text (this can include HTML)
- The footer text (this can include HTML)

In addition to these settings, you can configure the following using the Merchant Interface:

- Text: font family and font size
- Heading: font family, font size, font style and font color

To configure these settings in the Merchant Interface, log on to your merchant account. Click **Settings** in the **Account** section of the left-hand menu, then click **Payment Form** in the Transaction Format section. Click **Color and Font Settings** to open the color and font configuration page. Click **Help** to see complete instructions on how to use this page.

The following table describes the fields that can be submitted using the HTML Form POST to customize the merchant's payment form to look like their website.

The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the field">
```

Note: Fields submitted to the payment gateway by means of a transaction request override field settings established in the Merchant Interface.

FIELD NAME	VALUE	FORMAT	NOTES
x_header_html_payment_form	The hosted payment form header	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed as the header on the hosted payment form. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with the HTML Form POST. With this method there is no character limit.
x_footer_html_payment_form	The hosted payment form footer	Plain text or HTML Avoid using double quotes.	The text or HTML submitted in this field is displayed as the footer on the hosted payment form.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
			When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with HTML Form POST. With this method there is no character limit.
x_color_background	The background color	Any valid HTML color name or color hex code	The value in this field will set the background color for the hosted payment form and receipt page.
x_color_link	The hyperlink color	Any valid HTML color name or color hex code	The value in this field will set the color of the HTML links for the hosted payment form and receipt page.
x_color_text	The text color	Any valid HTML color name or color hex code	The value in this field will set the color of the text on the hosted payment form and the receipt page.
x_logo_url	The URL of the merchant's logo		The image referenced by this URL is displayed in the header or footer of the hosted payment form and receipt page. Logo images must be uploaded to the payment gateway server. See the "Logos and background images for the payment form" section of this guide for more information on how to upload images.
x_background_url	The URL of the merchant's background image		The image referenced by this URL is displayed as the background on the hosted payment form and receipt page. Background images must be uploaded to the payment gateway server. See the "Logos and background

FIELD NAME	VALUE	FORMAT	NOTES
			images for the payment form" section of this guide for more information on how to upload images.

IMPORTANT: All URLs referenced in the payment form header and footer such as links, images, and cascading style sheets **must** be absolute URLs. Also, please be aware that even though the hosted payment form is secure, the lock icon on the user's status bar might key off the location of the referenced file and not off the payment form. If the referenced file is not hosted on a secure server, the lock icon will turn off and the page will not look secure to the customer.

Using a cascading style sheet (CSS) with the hosted payment form

You can further customize the look of the payment gateway hosted payment form to match the text styles of the merchant's website by using a cascading style sheet in the header of the hosted payment form.

The payment form header can be configured in the Merchant Interface, or by submitting form fields with the HTML Form POST.

Note: The maximum character length allowed when configuring payment form header or footer texts in the Merchant Interface is 255. If you are declaring several styles in the payment form header or footer, it is recommended that you submit the style sheet in the payment form header or footer fields (*x_header_html_payment_form*, *x_footer_html_payment_form*) using the transaction request. With this method, there is no character limit.

The sample code below shows how to include a style sheet in the transaction request:

<div align="center">Sample 5. Including a style sheet in the HTML Form POST</div>

```
<INPUT TYPE=HIDDEN NAME="x_header_html_payment_form"
VALUE="<style type='text/css' media='all'>TD{font-family:
arial, verdana,trebuchet,helvetica,geneva,sans-serif;font-
size:11px; color:#000000;margin-left:1px;} INPUT{font-
family:Arial,Verdana, Trebuchet,Helvetica,Geneva,sans-
serif;font-size:11px;color: #000000;margin-
left:1px;}</style> Please enter your payment and shipping
information.">
```

The sample code below shows how to include a style sheet in the Merchant Interface payment form header text field:

Last revised: 8/30/2010

Sample 6. Including a style sheet in the Merchant Interface payment form header

```
<style type='text/css' media='all'>TD,input{font-
family:arial, verdana,trebuchet,helvetica,geneva,sans-
serif;font-size:11px; color:#000000;};h2{font-
family:arial,sans-serif;font-size:11px;
color:#000000;}</style>

<h2> Please enter your payment and shipping
information.</h2>
```

For more information about configuring the look and feel of the hosted payment form in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Logos and Background Images for the Hosted Payment Form

Merchants can request that their logos and/or background images be displayed on the hosted payment form. These requests can be done either online through the Merchant Interface, or by email.

To submit an image for hosting through the Merchant Interface:

1. Log on to the Merchant Interface at https://account.authorize.net.
2. Click **Contact Us** from the upper right corner of any Merchant Interface page.
3. Click **Create a New eTicket**.
4. Verify your contact information, enter your request in the space provided, and click **Submit**. The eTicket detail window opens.
5. In the Attachments section, click **Add**.
6. Click the check box to the right of the Attachment Name field. The Add Attachment window opens.
7. Click **Browse…**
8. Locate and select the image you wish to upload, and click **Open**.
9. Click **Add**.
10. Click **Submit**.

Your request will be sent to our Customer Support department. Please allow two business days for uploads to become available. Once we have hosted your image, we will send you specific instructions on how to reference the file. You can check the status of your eTicket at any time by logging on to the Merchant Interface, clicking the Contact Us link, and clicking Manage Existing eTickets.

Once Customer Support has responded to your request, you will see a yellow banner at the top of the Merchant Interface. Click View eTicket from the yellow banner to review the response from Customer Support. You will also receive an e-mail notification with a link to log into the Merchant Interface to review your eTicket.

To submit an image for hosting through e-mail:

*   Send an e-mail with your request, your Payment Gateway ID, and the image file as an attachment to: support@authorize.net.

Please allow two business days for uploads to become available. Once we have hosted your image, we will send you specific instructions on how to reference the file.

Last revised: 8/30/2010

Requirements and Guidelines

- Images must be in JPEG, GIF, or PNG formats. Other file formats will not be accepted.
- Please name the file using the convention `logo_GatewayID.ext`, where *GatewayID* is your Payment Gateway ID (up to six digits), and where *ext* is either `jpg`, `gif`, or `png`. For more information, please see the <u>knowledge base</u> article "What is my Payment Gateway ID."

If you have already submitted an image but have not received an update within two business days, please contact Customer Support so we can verify that we received the image and have submitted it for hosting.

We strongly recommend smaller files to ensure that your customers can view the full payment form quickly. On a 56K modem connection, a 1 Mb image can take up to 2.5 minutes to display, while faster connections may take just a few seconds.

The Authorize.Net hosted payment form is 580 pixels wide. Images wider that 580 pixels may not fit properly on the form's header or footer. Logos and background images can be wider than 580 pixels but we recommend keeping the image a reasonable size for Web hosting.

Images that are too tall may result in your customers needing to scroll down to reach the payment form. We recommend keeping the image a reasonable size for Web hosting.

Merchant-defined fields

Merchants can also choose to include merchant-defined fields to further customize the information included with a transaction. Merchant-defined fields are any fields that are not recognized by the payment gateway as standard application programming interface (API) payment form fields.

For example, the merchant might want to provide a field in which customers can provide specific shipping instructions and product color information. All you need to do is submit a custom field name and any accompanying text with the payment form request—for example, *shipping_instructions* and *product_color*.

Warning: Merchant-Defined Data fields are not intended to and MUST NOT be used to capture personally identifying information. Accordingly, Merchant is prohibited from capturing, obtaining, and/or transmitting any personally identifying information in or by means of the Merchant-Defined Data fields. Personally identifying information includes, but is not limited to, name, address, credit card number, social security number, driver's license number, state-issued identification number, passport number, and card verification numbers (CVV, CVC2, CVV2, CID, CVN). In the event that Authorize.Net, a CyberSource solution, discovers that Merchant is capturing and/or transmitting personally identifying information by means of the Merchant-Defined Data fields, whether or not intentionally, CyberSource WILL immediately suspend Merchant's account, which will result in a rejection of any and all transaction requests submitted by Merchant after the point of suspension.

The HTML sample code below is an example of how to submit merchant-defined fields. The result of this sample code is a page that displays the payment form button and the merchant-defined fields. Data submitted using merchant-defined fields is echoed back in merchant confirmation emails (see the "Email Receipt" section of this guide for more information).

Note: The sample code included in this document uses dummy field values in an ASP scripting environment. Because code varies based on the Web programming language used, it is not recommended that you copy and paste sample code but rather use it as a guide. Additional

Last revised: 8/30/2010

sample code is available for download from the Authorize.Net Developer Center at http://developer.authorize.net.

Sample 9. Payment form request with merchant-defined transaction data

```asp
<!--#INCLUDE FILE="simlib.asp"-->
<FORM METHOD=POST ACTION=
"https://secure.authorize.net/gateway/transact.dll">
   <% ret = InsertFP (APIloginid, sequence, amount,
txnkey) %>


<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the
merchant's API Login ID">
<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">
<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">
<INPUT TYPE=HIDDEN NAME="x_show_form"
VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_invoice_num" VALUE="ORDER-
002450">
<INPUT TYPE=HIDDEN NAME="x_description" VALUE="Product
or order description.">
<INPUT TYPE=HIDDEN NAME="x_cust_id" VALUE="Doe-John
001">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">
Enter any special shipping instructions:
<INPUT TYPE=TEXT NAME="shipping_instructions"><BR>
Enter desired product color:
<INPUT TYPE=TEXT NAME="product_color"><BR><BR>
<INPUT TYPE=SUBMIT VALUE="Click here for the secure
payment form">
</FORM>
```

Figure 4. The payment form button with merchant-defined fields

The *shipping_instructions* and *product_color* fields are not standard payment gateway fields, and are therefore treated as merchant-defined fields.

Last revised: 8/30/2010

Note: Standard payment gateway fields that are misspelled are treated as merchant-defined fields.

Renaming a Field

You can change the values on the Authorize.Net hosted payment form by using the `x_rename` field. Pass this as a hidden variable in your transaction request, and set it so it mentions the field you wish to rename and the new name, separated by a comma.

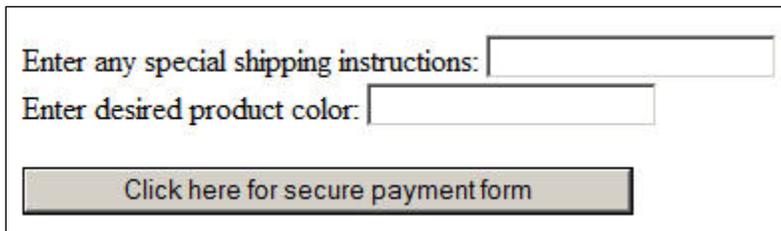For example, if you wish to replace the "Customer ID" field name on the payment form with the words "T-Shirt Size (S, M, L)", you can place the following in your code:

```
<input type="hidden" name="x_rename" value="x_cust_id,T-Shirt
Size (S, M, L)">
```

This will cause the words "T-Shirt Size (S, M, L)" to replace "Customer ID" on the payment form and in the email receipts.

Note that `x_rename` does not rename the original field when the transaction response is posted back to your server. It only renames the payment form field name. Using the above example, if the customer entered "L" in the renamed Customer ID field, the transaction response would include this field and value:

```
x_cust_id = "L"
```

Last revised: 8/30/2010

Section 4
Receipt Options

In addition to the secure payment form, SIM also provides two options for communicating the transaction results to the customer: the payment gateway hosted receipt page OR Relay Response.

The hosted receipt page is a brief transaction summary that is displayed in the customer's Web browser from the secure payment gateway server. It can be configured to match the look and feel of the merchant's website.

The Relay Response feature of SIM allows the merchant to create a custom receipt page using transaction results information returned by the payment gateway. The custom receipt page is then relayed to the customer's Web browser.

Note: You should implement only one receipt page option. Implementing both options can cause integration errors. Please consult with the merchant to determine which receipt option best meets their business needs.

In addition, the merchant can choose to send their customers the payment gateway automated email receipt.

The Receipt Page

The hosted receipt page provides the customer with the status of their transaction and can include a link back to the merchant's website. It can be customized to reflect the look and feel of the merchant's website.

Customizing the hosted receipt page

Settings for the hosted receipt page can be configured by passing fields in the transaction request on a per-transaction basis, or in the Merchant Interface.

Note: You might want to consider configuring this and other important integration settings using the HTML Form POST payment form request to prevent the integration from being affected if this and similar settings are inadvertently changed by the merchant in the Merchant Interface.

The following settings are necessary for the receipt page:

Receipt Link URL(s)

A receipt link URL can be displayed in the receipt page header and is used to redirect a customer from the hosted receipt page back to the merchant's website. To be accepted as valid by the payment gateway and to be displayed on the receipt page, the receipt link URL submitted in the transaction request must also be configured in the Merchant Interface.

Receipt Method

This setting specifies the kind of link that is made back to the merchant's website.

- LINK creates a regular hyperlink.
- GET creates a button and returns transaction information in the receipt link URL.
- POST creates a button and returns transaction information as an HTML Form POST.

For more information on configuring these settings in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

The following table describes the form fields that can be submitted to customize the hosted receipt page. The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" "VALUE="value of the field">
```

FIELD NAME	VALUE	FORMAT	NOTES
x_receipt_link_method	The type of link back to the merchant's website from the hosted receipt page	LINK, POST, or GET	LINK creates a regular hyperlink. GET creates a button and returns transaction information in the receipt link URL. POST creates a button and returns transaction information as an HTML Form POST.
x_receipt_link_text	The text of the link or button that directs the customer back to the merchant's website	Up to 50 characters	If the x_receipt_link_method is LINK, the value in this field is a hyperlinked text on the hosted receipt page. If the x_receipt_link_method is GET or POST, the value in this field becomes the text of a submit button. An HTML form is created in the receipt page that has hidden fields containing the results of the transaction processed.
x_receipt_link_url	The URL of the link or button that directs the customer back to the		To be accepted as valid by the payment gateway, the receipt link URL must also be configured in the

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
	merchant's website		Merchant Interface. If the x_receipt_link_method is LINK, the URL specified becomes the href value of the hyperlinked text. If the x_receipt_link_method is GET or POST, the URL will become the action of the HTML form.

The following sample code is an example of including a receipt link for the hosted receipt page in the transaction request.

Note: The sample code included in this document uses dummy field values in an ASP scripting environment. Because code varies based on the Web programming language used, it is not recommended that you copy and paste sample code but rather use it as a guide. Additional sample code is available for download from the Authorize.Net Developer Center at http://developer.authorize.net.

Sample 10. Payment form request including receipt link URL

```
<!--#INCLUDE FILE="simlib.asp"-->

<FORM METHOD=POST ACTION=

"https://secure.authorize.net/gateway/transact.dll">

    <% ret = InsertFP (APIloginid, sequence, amount,
txnkey) %>

<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the
merchant's API Login ID">

<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">

<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">

<INPUT TYPE=HIDDEN NAME="x_show_form"
VALUE="PAYMENT_FORM">

<INPUT TYPE=HIDDEN NAME="x_invoice_num" VALUE="ORDER-
002450">

<INPUT TYPE=HIDDEN NAME="x_description" VALUE="Product
or order description.">

<INPUT TYPE=HIDDEN NAME="x_cust_id" VALUE="Doe-John
001">

<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">

<INPUT TYPE=HIDDEN NAME="x_receipt_link_method"
VALUE="LINK">

<INPUT TYPE=HIDDEN NAME="x_receipt_link_text"
VALUE="Return to our online store">
```

```
<INPUT TYPE=HIDDEN NAME="x_receipt_link_URL"
VALUE="http://www.mydomain.com">

<INPUT TYPE=SUBMIT VALUE="Click here for the secure
payment form">

</FORM>
```

The sample code above produces a button that is displayed to the customer upon checkout (see Figure 1. in the "Submitting Transactions" section of this guide). When the customer clicks the button, the secure hosted payment form is displayed (see Figure 2. in the "Submitting Transactions" section of this guide).

Once the customer submits the transaction, the hosted receipt page (Figure 5. below) is displayed.

Figure 5. The hosted receipt page with receipt link URL

IMPORTANT: Submitting these fields will provide basic placement of the receipt link URL on the receipt page. To customize the placement of a URL on the receipt page, it can be referenced in HTML in either the receipt page header or footer API fields

Last revised: 8/30/2010

(x_header_html_receipt, x_footer_html_receipt) or in the Merchant Interface receipt page header and footer settings.

Customizing the receipt page look and feel

When using the hosted receipt page, the following settings can be configured to match the look of the merchant's website.

- The color of the text
- The color of the link text
- The background color
- The header text (this can include HTML)
- The footer text (this can include HTML)

In addition to these settings, you can configure the following using the Merchant Interface:

- Text: font family and font size
- Heading: font family, font size, font style and font color

To configure these settings in the Merchant Interface, log on to your merchant account. Click **Settings** in the **Account** section of the left-hand menu, then click **Receipt Page** in the Transaction Submission section. Click **Color and Font Settings** to open the color and font configuration page. Click **Help** to see complete instructions on how to use this page.

The following table describes the fields that can be submitted by means of the HTML Form POST to customize the merchant's payment form to look like their website.

The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the field">
```

Note: Fields submitted to the payment gateway by means of a transaction request override field settings established in the Merchant Interface.

FIELD NAME	VALUE	FORMAT	NOTES
x_header_html_receipt	The hosted receipt page header	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed at the top of the hosted receipt page. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with the HTML Form POST on a per-transaction basis. With this method there is no character limit.
x_footer_html_receipt	The hosted receipt page footer	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed at the bottom of the hosted receipt page. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
			submit this field with the HTML Form POST on a per-transaction basis. The length for this field in the Merchant Interface is limited to 255 characters.
x_color_background	The background color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the background color for both.
x_color_link	The hyperlink color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the color of the HTML links for both.
x_color_text	The text color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the color of the text on the hosted payment form and the receipt page.
x_logo_url	The URL of the merchant's logo		The image referenced by this URL is displayed on the header of the hosted payment form and receipt page. Logo images must be uploaded to the payment gateway server. See the "Logos and background images for the receipt page" section of this guide for more information on how to upload images.
x_background_url	The URL of the merchant's background image		The image referenced by this URL is displayed as the background of the hosted payment form and receipt page. Background images must be uploaded to the payment gateway server. See the "Logos and background images for the receipt page" section of this guide for more information on how to upload images.

IMPORTANT: All URLs referenced in the receipt page header and footer such as links, images, and cascading style sheets **must** be absolute URLs. Even though the hosted receipt page is secure, the lock icon on the user's task bar might key off the location of the referenced file and not off the receipt page.

Last revised: 8/30/2010

Using a cascading style sheet (CSS) with the hosted receipt page

You can further customize the look of the payment gateway hosted receipt page to match the text styles of the merchant's website by using a cascading style sheet in the header or footer of the hosted receipt page.

The receipt page header and footer can be configured in the Merchant Interface, or by submitting form fields with the HTML Form POST.

Note: The maximum character length allowed when configuring receipt page header or footer texts in the Merchant Interface is 255. If you are declaring several styles in the receipt page header or footer, it is recommended that you submit the style sheet in the receipt page header or footer fields (*x_header_html_receipt*, *x_footer_html_receipt*) using the HTML form POST. With this method, there is no character limit.

The sample code below shows how to include a style sheet in the HTML Form POST:

Sample 11. Including a style sheet in the HTML Form POST

```
<INPUT TYPE=HIDDEN NAME="x_header_html_receipt"
VALUE="<style type='text/css' media='all'>TD{font-family:
arial, verdana,trebuchet,helvetica,geneva,sans-serif;font-
size:11px; color:#000000;margin-left:1px;}">
```

The sample code below shows how to include a style sheet in the Merchant Interface receipt page header field:

Sample 12. Including a style sheet in the Merchant Interface receipt page header

```
<style type='text/css' media='all'>TD,input{font-
family:arial, verdana,trebuchet,helvetica,geneva,sans-
serif;font-size:11px; color:#000000;};h2{font-
family:arial,sans-serif;font-size:11px;
color:#000000;}</style>
```

For more information about configuring the look and feel of the hosted receipt page in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Logos and background images for the hosted receipt page

If the merchant is using a logo and/or background images on the hosted payment form, the same image files can be referenced for display on the hosted receipt page. Image files must be uploaded to the payment gateway server in order to be displayed properly. For more information on how to upload image files, please see the "Logos and Background Images for the Hosted Payment Form" section of this guide.

IMPORTANT: Submitting the *x_logo_url* and *x_background_url* fields will provide basic placement of the logo and background images on the receipt page. To customize the placement of these images on the receipt page, they can be referenced in HTML in either

the receipt page header or footer fields (*x_header_html_receipt* and *x_footer_html_receipt*) or in the Merchant Interface receipt page header and footer settings.

Relay Response

Relay Response does not redirect the end user back to your server, but relays your specified Relay URL to the end user through our receipt page instead of displaying our default receipt page. If you would like to redirect the end user back to your server, please provide a link on your Relay URL for this purpose.

The following table describes form fields that can be submitted to configure Relay Response. These settings can also be configured in the Merchant Interface. For more information about configuring Relay Response in the Merchant Interface, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the
field">
```

FIELD NAME	VALUE	FORMAT	NOTES
x_relay_response	The request for a relay response	TRUE	This field instructs the payment gateway to return transaction results to the merchant by means of an HTML form POST to the merchant's Web server for a relay response.
x_relay_url	The URL on the merchant's website to which the payment gateway posts transaction results for a relay response	Any valid URL Including name/value pairs in the URL(anything after a "?") is not recommended	In the event that this field is submitted, the payment gateway will validate the URL value against the Relay Response URL configured in the Merchant Interface. If the URL submitted does not match the URL configured in the Merchant Interface, the transaction will be rejected. If no value is submitted in the HTML Form POST, the payment gateway will post transaction results to the URL configured in the Merchant Interface.

Note: If the merchant would like to use the payment gateway hosted receipt page, the Relay Response fields listed above should not be submitted in the transaction request, nor should they be configured in the Merchant Interface. Requesting both the hosted receipt page and a Relay Response will result in a failed implementation.

The following sample code is an example of including the Relay Response request in the HTML Form POST.

Note: The sample code included in this document uses dummy field values in an ASP scripting environment. Because code varies based on Web programming language, it is not recommended that you copy and paste sample code but rather use it as a guide. Additional

Last revised: 8/30/2010

sample code is available for download from the Authorize.Net Developer Center at
http://developer.authorize.net.

Sample 13. Payment form request including Relay Response request

```
<!--#INCLUDE FILE= "simlib.asp"-->

<FORM METHOD=POST ACTION=

"https://secure.authorize.net/gateway/transact.dll">

   <% ret = InsertFP (APIloginid, sequence, amount,
txnkey) %>

<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the
merchant's API Login ID">

<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">

<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">

<INPUT TYPE=HIDDEN NAME="x_show_form"
VALUE="PAYMENT_FORM">

<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">

<INPUT TYPE=HIDDEN NAME="x_relay_response"
VALUE="TRUE">

<INPUT TYPE=HIDDEN NAME="x_relay_url" VALUE="Any valid
URL">

<INPUT TYPE=SUBMIT VALUE="Click here for the secure
payment form">

</FORM>
```

When Authorize.Net sends a Relay Response to the merchant's server, if the merchant's web server does not send a positive response after ten seconds, then the connection will time-out and an error is generated for the transaction.

If the merchant would like to use transaction response information for purposes other than creating a custom receipt page, such as integration with proprietary business processes or applications, a Silent Post URL can be configured in the Merchant Interface. The Silent Post URL is a location on the merchant's Web server where the payment gateway can "carbon copy" the transaction information. This allows the merchant to use transaction information for other purposes separately without affecting the amount of time it takes to respond to the payment gateway with a custom receipt page. Note that Relay Response is the recommended method of collecting transaction response data.

Note: The Silent Post feature can only be configured in the Merchant Interface. There is no API field for the Silent Post URL. For information on how to configure the Silent Post URL in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

The Silent Post URL does *not* function the same as the Relay Response URL. Once the payment gateway posts the transaction information to the Silent Post URL, the communication is complete. To receive payment or event notification, use Relay Response, because Silent Posts are neither logged nor monitored. The receipt of Silent Post responses are not tracked, and there is no notification in case of a failed receipt of Silent Post responses.

Last revised: 8/30/2010

Note:	Authorize.Net only posts the Silent Post on ports 80 and 443.

Tips for using Relay Response:

The Relay Response URL specified should be a script that can parse the transaction results posted from the payment gateway. The URL can be a plain HTML page if a static response is desired for every transaction. However, in this case the merchant's Web server should be configured to allow an HTML Form POST to a plain HTML page.

The HTTP header should not be relied on for including customer information such as cookies. When the response is relayed to the customer's browser, HTTP headers are replaced.

Keep in mind that the relay response is rendered on the payment gateway server. Custom receipt pages **must** incorporate absolute URLs.

Redirects or frames in the relay script are not recommended, because the information might not be transferred properly.

Email Receipt

Merchants can choose to send a payment gateway-generated email receipt to customers who provide an email address with their transaction. The email receipt includes a summary and results of the transaction. To the customer, this email appears to be sent from the merchant contact that is configured as the Email Sender in the Merchant Interface. (For more information about the Email Sender setting, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.)

To send the payment gateway-generated customer email receipt, the following API fields can be submitted with the transaction request string. These settings can also be configured in the Merchant Interface. For more information about configuring these settings in the Merchant Interface, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the field">
```

FIELD NAME	VALUE	FORMAT	NOTES
x_email	The customer's valid email address	Up to 255 characters Example: janedoe@customer.com	The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid. For more information about Email Receipts, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_email_customer	The customer	TRUE, FALSE,	Indicates whether an email

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
	email receipt status	T, F, YES, NO, Y, N, 1, 0	receipt should be sent to the customer.
			If set to TRUE, the payment gateway will send an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer.
			If no value is submitted, the payment gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has enabled the setting. If this field is not submitted and the setting is disabled in the Merchant Interface, no email is sent.
			For more information about configuring Email Receipts in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_header_email_receipt	The email receipt header	Plain text	This text will appear as the header of the email receipt sent to the customer.
x_footer_email_receipt	The email receipt footer	Plain text	This text will appear as the footer on the email receipt sent to the customer.

In addition, the merchant can receive a transaction confirmation email from the payment gateway at the completion of each transaction, which includes order information and the results of the transaction. Merchants can sign up for confirmation emails in the Merchant Interface. For more information, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Section 5
Additional API Fields

The following tables describe additional API fields that can be submitted by means of the transaction request to the payment gateway. The form fields are submitted using the syntax:

```
<INPUT TYPE=HIDDEN NAME="x_name_of_field" VALUE="value of the
field">
```

Transaction Information

The following fields contain optional or conditional transaction-specific information.

FIELD NAME	VALUE	FORMAT	NOTES
x_version	The merchant's transaction version	3.0, 3.1	Indicates to the system the set of fields that will be included in the response: 3.0 is the default version. 3.1 allows the merchant to utilize the Card Code feature, and is the current standard version. It is highly recommended that you submit this field on a per-transaction basis, particularly if you are using Relay Response. For more information, see the "Relay Response" and "Appendix A Fields by Transaction Type" sections of this document.
x_method	The payment method	CC or ECHECK	The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If left blank, this value defaults to CC. For more information about eCheck.Net transaction requirements, see the *eCheck.Net Developer Guide* at http://developer.authorize.net/guides/echeck.pdf.
x_test_request	The request to process test transactions	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates if the transaction should be processed as a test transaction. See the "Test Transactions" section of this guide for more information.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
x_duplicate_window	The window of time after the submission of a transaction that a duplicate transaction can not be submitted	Any value between 0 and 28800 (no comma)	Indicates in seconds the window of time after a transaction is submitted during which the payment gateway will check for a duplicate transaction. The maximum time allowed is 8 hours (28800 seconds). If a value less than 0 is sent, the payment gateway will default to 0 seconds. If a value greater than 28800 is sent, the payment gateway will default to 28800. If no value is sent, the payment gateway will default to 2 minutes (120 seconds). If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See the "Response for Duplicate Transactions" section of this guide for more information.

Itemized Order Information

Based on their respective business requirements, merchants can choose to submit itemized order information with a transaction. Itemized order information is not submitted to the processor and is currently not returned with the transaction response. This information is displayed on the Transaction Detail page and in the QuickBooks download file reports in the Merchant Interface.

Note: The value for the *x_line_item* field is capable of including delimited item information. In this case, line item values must be included in the order listed below.

FIELD NAME	VALUE	FORMAT	NOTES
x_line_item	Any string	Line item values must be delimited by a bracketed pipe <|>	Itemized order information.
	Item ID<|>	Up to 31 characters	The ID assigned to an item.
	<|>item name<|>	Up to 31 characters	A short description of an item.
	<|>item description<|>	Up to 255 characters	A detailed description of an item.
	<|>itemX quantity<|>	Up to two decimal places	The quantity of an item.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
		Must be a positive number	
	<|>item price (unit cost)<|>	Up to two decimal places Must be a positive number	Cost of an item per unit, *excluding* tax, freight, and duty. The dollar sign ($) is not allowed when submitting delimited information.
	<|>itemX taxable	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates whether the item is subject to tax.

The merchant can submit up to 30 distinct line items containing itemized order information per transaction. For example:

Sample 14. Submitting itemized order information

```
<INPUT TYPE="HIDDEN" name="x_line_item"
    VALUE="item1<|>golf balls<|><|>2<|>18.95<|>Y">
<INPUT TYPE="HIDDEN" name="x_line_item"
    VALUE="item2<|>golf bag<|>Wilson golf carry bag,
    red<|>1<|>39.99<|>Y">
<INPUT TYPE="HIDDEN" name="x_line_item"
    VALUE="item3<|>book
<|>Golf for Dummies<|>1<|>21.99<|>Y">
```

Note: For Prior Authorization and Capture transactions, if line item information was submitted with the original transaction, adjusted information can be submitted in the event that the transaction changed. If no adjusted line item information is submitted, the information submitted with the original transaction will apply.

Additional Customer Information

The following fields describe additional customer information that can be submitted with each transaction.

FIELD NAME	VALUE	FORMAT	NOTES

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
x_customer_ip	The customer's IP address	Up to 15 characters (no letters) Example: 255.255.255.255	IP address of the customer initiating the transaction. If this value is not passed, it will default to 255.255.255.255. This field is required when using the Fraud Detection Suite™ (FDS) IP Address Blocking tool. For more information about FDS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Section 6
Transaction Response

When Relay Response is configured, the transaction response that is returned to the merchant from the payment gateway is a set of fields that provides information about the status of a transaction—whether it was accepted or declined—as well as information included in the transaction request.

The merchant server can parse data in the transaction response and customize the message to display to the customer. Transaction results are also provided in the merchant confirmation email, customer email receipt (if configured), and on the Transaction Detail page for the transaction in the Merchant Interface.

Fields in the Payment Gateway Response

The following table lists the fields returned in the response from the payment gateway.

FIELD NAME	VALUE	FORMAT	NOTES
x_response_code	The overall status of the transaction	1 = Approved 2 = Declined 3 = Error 4 = Held for Review	
x_response_subcode	A code used by the payment gateway for internal transaction tracking		
x_response_reason_code	A code that represents more details about the result of the transaction	Numeric	See the "Response Code Details" section of this document for a listing of response reason codes.
x_response_reason_text	A brief description of the result, which corresponds with the response reason code	Text	You can generally use this text to display a transaction result or error to the customer. However, please review the "Response Code Details" section of this document to identify any specific texts you do not

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
			want to pass to the customer.
x_auth_code	The authorization or approval code	6 characters	
x_avs_code	The Address Verification Service (AVS) response code	A = Address (Street) matches, ZIP does not B = Address information not provided for AVS check E = AVS error G = Non-U.S. Card Issuing Bank N = No Match on Address (Street) or ZIP P = AVS not applicable for this transaction R = Retry – System unavailable or timed out S = Service not supported by issuer U = Address information is unavailable W = Nine digit ZIP matches, Address (Street) does not X = Address (Street) and nine digit ZIP match Y = Address (Street) and five digit ZIP match Z = Five digit ZIP matches, Address (Street) does not	Indicates the result of the (AVS) filter. For more information about AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_trans_id	The payment gateway assigned identification number for the transaction	When x_test_request is submitted, this value will be "0."	This value can be used to perform related follow-up transactions using AIM. This value is also used in the MD5 Hash value.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
x_invoice_num	The merchant assigned invoice number for the transaction	Up to 20 characters (no symbols)	
x_description	The transaction description	Up to 255 characters (no symbols)	
x_amount	The amount of the transaction	Up to 15 digits	
x_method	The payment method	CC or ECHECK	
x_type	The type of credit card transaction	AUTH_CAPTURE, AUTH_ONLY, CREDIT, PRIOR_ AUTH_CAPTURE, VOID	
x_cust_id	The merchant assigned customer ID	Up to 20 characters (no symbols)	
x_first_name	The first name associated with the customer's billing address	Up to 50 characters (no symbols)	
x_last_name	The last name associated with the customer's billing address	Up to 50 characters (no symbols)	
x_company	The company associated with the customer's billing address	Up to 50 characters (no symbols)	
x_address	The customer's billing address	Up to 60 characters (no symbols)	
x_city	The city of the customer's billing address	Up to 40 characters (no symbols)	
x_state	The state of the customer's	Up to 40 characters (no symbols) or a valid two-	

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
	billing address	character state code	
x_zip	The ZIP code of the customer's billing address	Up to 20 characters (no symbols)	
x_country	The country of the customer's billing address	Up to 60 characters (no symbols)	
x_phone	The phone number associated with the customer's billing address	Up to 25 digits (no letters) Ex. (123)123-1234	
x_fax	The fax number associated with the customer's billing address	Up to 25 digits (no letters) Ex. (123)123-1234	
x_email	The customer's valid email address	Up to 255 characters	
x_ship_to_first_name	The first name associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_last_name	The last name associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_company	The company associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_address	The customer's shipping address	Up to 60 characters (no symbols)	
x_ship_to_city	The city of the customer's shipping address	Up to 40 characters (no symbols)	
x_ship_to_state	The state of the customer's shipping address	Up to 40 characters (no symbols) or a valid two-character state code	

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
x_ship_to_zip	The ZIP code of the customer's shipping address	Up to 20 characters (no symbols)	
x_ship_to_country	The country of the customer's shipping address	Up to 60 characters (no symbols)	
x_tax	The tax amount charged	Numeric	Delimited tax information is not included in the transaction response.
x_duty	The duty amount charged	Numeric	Delimited duty information is not included in the transaction response.
x_freight	The freight amount charged	Numeric	Delimited freight information is not included in the transaction response.
x_tax_exempt	The tax exempt status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	
x_po_num	The merchant assigned purchase order number	Up to 25 characters (no symbols)	
x_MD5_Hash	The payment gateway generated MD5 hash value that can be used to authenticate the transaction response.		For more information about creating an MD5 hash value, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ .
x_cvv2_resp_code	The card code verification (CCV) response code	M = Match N = No Match P = Not Processed S = Should have been present U = Issuer unable to process request	Indicates the result of the CCV filter. For more information about CCV, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Last revised: 8/30/2010

FIELD NAME	VALUE	FORMAT	NOTES
x_cavv_response	The cardholder authentication verification response code	Blank or not present = CAVV not validated 0 = CAVV not validated because erroneous data was submitted 1 = CAVV failed validation 2 = CAVV passed validation 3 = CAVV validation could not be performed; issuer attempt incomplete 4 = CAVV validation could not be performed; issuer system error 5 = Reserved for future use 6 = Reserved for future use 7 = CAVV attempt – failed validation – issuer available (U.S.-issued card/non-U.S acquirer) 8 = CAVV attempt – passed validation – issuer available (U.S.-issued card/non-U.S. acquirer) 9 = CAVV attempt – failed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer) A = CAVV attempt – passed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer) B = CAVV passed validation, information	The cardholder authentication programs are not applicable to SIM.

FIELD NAME	VALUE	FORMAT	NOTES
		only, no liability shift	

Using the MD5 Hash Feature

The MD5 Hash feature enables you to authenticate that a transaction response is securely received from Authorize.Net. The payment gateway creates the MD5 hash using the following pieces of account and transaction information as input:

- MD5 Hash value
- API Login ID (*x_login*)
- Transaction ID (*x_trans_id*)
- Amount (*x_amount*)

The MD5 Hash value is a random value configured by the merchant in the Merchant Interface. It should be stored securely separately from the merchant's Web server. For more information on how to configure this value, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

For example, if the MD5 Hash value configured by the merchant in the Merchant Interface is `wilson`, the API Login ID is `myAPIloginid`, the Transaction ID is `987654321`, and the amount is $1.00, then the field order used by the payment gateway to generate the MD5 Hash would be as follows.

Sample 15. MD5 Hash input field order

```
wilsonmyAPIloginid9876543211.00
```

Note: The value passed back for *x_amount* is formatted with the correct number of decimal places used in the transaction. For transaction types that do not include a transaction amount, mainly Voids, the amount used by the payment gateway to calculate the MD5 Hash is "0.00."

To authenticate the MD5 Hash returned by the payment gateway in the transaction response, you need to create a script that can receive and parse the transaction response, call the merchant's MD5 Hash value, and run the MD5 algorithm on the same fields listed above. If the result matches the MD5 hash returned by the payment gateway, the transaction response is successfully authenticated.

Response for Duplicate Transactions

The SIM API allows you to specify the window of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction (based on credit card number, invoice number, amount, billing address information, transaction type, etc.) using the duplicate window field (*x_duplicate_window*). The value for this field can be between 0 and 28800 seconds (maximum of 8 hours).

In the event that the transaction request does not include the duplicate window field, and the payment gateway detects a duplicate transaction within the default window of 2 minutes, the payment gateway response will contain the response code of 3 (processing error) with a response reason code of 11 (duplicate transaction) with no additional details.

Last revised: 8/30/2010

In the event that the transaction request *does* include the duplicate window field and value, and the payment gateway detects a duplicate transaction within the window of time specified, the payment gateway response for the duplicate transaction will include the response code and response reason code listed above, as well as information about the original transaction (as outlined below).

If the original transaction was declined, and a value was passed in the duplicate window field, the payment gateway response for the duplicate transaction will include the following information for the original transaction:

- The AVS result
- The CCV result
- The transaction ID
- The MD5 hash (if this feature was used for the original transaction)

If the original transaction was approved, and a value was passed in the duplicate window field, the payment gateway response will also include the authorization code for the original transaction. All duplicate transactions submitted after the duplicate window, whether specified in the transaction request or after the payment gateway's default 2 minute duplicate window, are processed normally.

SIM Relay Response

The response from the gateway to a request by means of SIM for a Relay Response consists of a set of fields returned as a POST string to the merchant server at the location indicated in the *x_relay_url* field.

SIM Transaction Response Versions

There are two versions of the response string. The set of fields in the response differ based on the response version.

Version 3.0

The version 3.0 response contains system fields from position 1 to 38 and echoes merchant defined fields from 39 on, in the order received by the system. Version 3.0 is the Payment Gateway default.

Version 3.1

The version 3.1 response string contains 68 system fields, with field number 39 representing the Card Code (CVV2/CVC2/CID) response code. Merchant-defined fields are echoed from field 69 on. Merchants wishing to use the Card Code feature must use transaction version 3.1.

Upgrading the Transaction Version

To upgrade the transaction version, do the following (only users with the appropriate permissions will be able to access this setting):

1. Log on to the Merchant Interface

2. Select **Settings** from the Main Menu

3. Click **Transaction Version** in the Transaction Response section

4. Change the transaction version using the drop-down box

5. Click **Submit**

Note: You can only upgrade to a higher transaction version. You cannot set your transaction
version to a previous version.

Response Code Details

The following tables describe the response codes and response reason texts that are returned for
each transaction. In addition to the information in this document, the Authorize.Net Developer
Center at http://developer.authorize.net/tools/responsereasoncode provides a valuable tool for
troubleshooting errors.

- **Response Code** indicates the overall status of the transaction with possible values of
  approved, declined, error, or held for review.
- **Response Reason Code** is a numeric representation of a more specific reason for the
  transaction status.
- **Response Reason Text** details the specific reason for the transaction status. This
  information can be returned to the merchant and/or customer to provide more information
  about the status of the transaction.

Response Codes

RESPONSE CODE	DESCRIPTION
1	This transaction has been approved.
2	This transaction has been declined.
3	There has been an error processing this transaction.
4	This transaction is being held for review.

Response Reason Codes and Response Reason Text

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
1	1	This transaction has been approved.	
2	2	This transaction has been declined.	
2	3	This transaction has been declined.	
2	4	This transaction has been declined.	The code returned from the processor indicating that the card used needs to be picked up.
3	5	A valid amount is required.	The value submitted in the amount field did not pass validation for a number.
3	6	The credit card number is invalid.	

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
3	7	The credit card expiration date is invalid.	The format of the date submitted was incorrect.
3	8	The credit card has expired.	
3	9	The ABA code is invalid.	The value submitted in the x_bank_aba_code field did not pass validation or was not for a valid financial institution.
3	10	The account number is invalid.	The value submitted in the x_bank_acct_num field did not pass validation.
3	11	A duplicate transaction has been submitted.	A transaction with identical amount payment information was submitted during the Duplicate Transaction Window for the original transaction. Please see the "Response for Duplicate Transactions" topic for more details.
3	12	An authorization code is required but not present.	A transaction that required x_auth_code to be present was submitted without a value.
3	13	The merchant API Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applicable only to SIM and WebLink APIs.
3	15	The transaction ID is invalid.	The transaction ID value is non-numeric or was not present for a transaction that requires it.
3	16	The transaction was not found.	The transaction ID sent in was properly formatted but the gateway had no record of the transaction for the gateway account used.
3	17	The merchant does not accept this type of credit card.	The merchant was not configured to accept the credit card type submitted in the transaction.
3	18	ACH transactions are not accepted by this merchant.	The merchant does not accept electronic checks.

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
3	19 - 23	An error occurred during processing. Please try again in 5 minutes.	
3	24	The Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
3	25 - 26	An error occurred during processing. Please try again in 5 minutes.	
2	27	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	
2	28	The merchant does not accept this type of credit card.	The Merchant ID at the processor was not configured to accept this card type.
2	29	The Paymentech identification numbers are incorrect. Call Merchant Service Provider.	
2	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
2	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	32	This reason code is reserved or not applicable to this API.	
3	33	*FIELD* cannot be left blank.	The word *FIELD* will be replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in. Please see the Form Settings topic in the Merchant Integration Guide for details.
2	34	The VITAL identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	35	An error occurred during processing. Call Merchant Service	The merchant was incorrectly set up at the processor.

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		Provider.	
3	36	The authorization was approved, but settlement failed.	
2	37	The credit card number is invalid.	
2	38	The Global Payment System identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	40	This transaction must be encrypted.	
2	41	This transaction has been declined.	Only merchants set up for the FraudScreen.Net service would receive this decline. This code will be returned if a given transaction's fraud score is higher than the threshold set by the merchant.
3	43	The merchant was incorrectly set up at the processor. Call your Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	44	This transaction has been declined.	The card code submitted with the transaction did not match the card code on file at the card issuing bank and the transaction was declined.
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	This occurs if the merchant tries to capture funds greater than the amount of the original authorization-only transaction.
3	48	This processor does not accept partial reversals.	The merchant attempted to settle for less than the originally authorized amount.
3	49	A transaction amount greater than	The transaction amount submitted was greater than the maximum

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		$[amount] will not be accepted.	amount allowed.
3	50	This transaction is awaiting settlement and cannot be refunded.	Credits or refunds can only be performed against settled transactions. The transaction against which the credit/refund was submitted has not been settled, so a credit cannot be issued.
3	51	The sum of all credits against this transaction is greater than the original transaction amount.	
3	52	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	
3	53	The transaction type was invalid for ACH transactions.	If x_method = ECHECK, x_type cannot be set to CAPTURE_ONLY.
3	54	The referenced transaction does not meet the criteria for issuing a credit.	
3	55	The sum of credits against the referenced transaction would exceed the original debit amount.	The transaction is rejected if the sum of this credit and prior credits exceeds the original debit amount
3	56	This merchant accepts ACH transactions only; no credit card transactions are accepted.	The merchant processes eCheck.Net transactions only and does not accept credit cards.
3	57 - 63	An error occurred in processing. Please try again in 5 minutes.	
2	65	This transaction has been declined.	The transaction was declined because the merchant configured their account through the Merchant Interface to reject transactions with certain values for a Card Code mismatch.
3	66	This transaction cannot be accepted for processing.	The transaction did not meet gateway security guidelines.
3	68	The version parameter is invalid.	The value submitted in x_version was invalid.
3	69	The transaction type is invalid.	The value submitted in x_type was invalid.

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
3	70	The transaction method is invalid.	The value submitted in x_method was invalid.
3	71	The bank account type is invalid.	The value submitted in x_bank_acct_type was invalid.
3	72	The authorization code is invalid.	The value submitted in x_auth_code was more than six characters in length.
3	73	The driver's license date of birth is invalid.	The format of the value submitted in x_drivers_license_dob was invalid.
3	74	The duty amount is invalid.	The value submitted in x_duty failed format validation.
3	75	The freight amount is invalid.	The value submitted in x_freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in x_tax failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in x_customer_tax_id failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in x_card_code failed format validation.
3	79	The driver's license number is invalid.	The value submitted in x_drivers_license_num failed format validation.
3	80	The driver's license state is invalid.	The value submitted in x_drivers_license_state failed format validation.
3	81	The requested form type is invalid.	The merchant requested an integration method not compatible with the AIM API.
3	82	Scripts are only supported in version 2.5.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	83	The requested script is either invalid or no longer supported.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	84	This reason code is reserved or not	

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		applicable to this API.	
3	85	This reason code is reserved or not applicable to this API.	
3	86	This reason code is reserved or not applicable to this API.	
3	87	This reason code is reserved or not applicable to this API.	
3	88	This reason code is reserved or not applicable to this API.	
3	89	This reason code is reserved or not applicable to this API.	
3	90	This reason code is reserved or not applicable to this API.	
3	91	Version 2.5 is no longer supported.	
3	92	The gateway no longer supports the requested method of integration.	
3	97	This transaction cannot be accepted.	Applicable only to SIM API. Fingerprints are only valid for a short period of time. This code indicates that the transaction fingerprint has expired.
3	98	This transaction cannot be accepted.	Applicable only to SIM API. The transaction fingerprint has already been used.
3	99	This transaction cannot be accepted.	Applicable only to SIM API. The server-generated fingerprint does not match the merchant-specified fingerprint in the x_fp_hash field.
3	100	The eCheck.Net type is invalid.	Applicable only to eCheck.Net. The value specified in the x_echeck_type field is invalid.
3	101	The given name on the account and/or the account type does not match the actual account.	Applicable only to eCheck.Net. The specified name on the account and/or the account type do not match the NOC record for this account.
3	102	This request cannot be accepted.	A password or Transaction Key

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
			was submitted with this WebLink request. This is a high security risk.
3	103	This transaction cannot be accepted.	A valid fingerprint, Transaction Key, or password is required for this transaction.
3	104	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for country failed validation.
3	105	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for city and country failed validation.
3	106	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for company failed validation.
3	107	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name does not contain valid characters.
3	120	An error occurred during processing. Please try again.	The system-generated void for the original timed-out transaction failed. (The original transaction timed out while waiting for a response from the authorizer.)
3	121	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a database error.)
3	122	An error occurred during	The system-generated void for the

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		processing. Please try again.	original errored transaction failed. (The original transaction experienced a processing error.)
3	123	This account has not been given the permission(s) required for this request.	The transaction request must include the API Login ID associated with the payment gateway account.
2	127	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	The system-generated void for the original AVS-rejected transaction failed.
3	128	This transaction cannot be processed.	The customer's financial institution does not currently allow transactions for this account.
3	130	This payment gateway account has been closed.	IFT: The payment gateway account status is Blacklisted.
3	131	This transaction cannot be accepted at this time.	IFT: The payment gateway account status is Suspended-STA.
3	132	This transaction cannot be accepted at this time.	IFT: The payment gateway account status is Suspended-Blacklist.
2	141	This transaction has been declined.	The system-generated void for the original FraudScreen-rejected transaction failed.
2	145	This transaction has been declined.	The system-generated void for the original card code-rejected and AVS-rejected transaction failed.
3	152	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	The system-generated void for the original transaction failed. The response for the original transaction could not be communicated to the client.
2	165	This transaction has been declined.	The system-generated void for the original card code-rejected transaction failed.
3	170	An error occurred during processing. Please contact the merchant.	Concord EFS – Provisioning at the processor has not been completed.

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
2	171	An error occurred during processing. Please contact the merchant.	Concord EFS – This request is invalid.
2	172	An error occurred during processing. Please contact the merchant.	Concord EFS – The store ID is invalid.
3	173	An error occurred during processing. Please contact the merchant.	Concord EFS – The store key is invalid.
2	174	The transaction type is invalid. Please contact the merchant.	Concord EFS – This transaction type is not accepted by the processor.
3	175	The processor does not allow voiding of credits.	Concord EFS – This transaction is not allowed. The Concord EFS processing platform does not support voiding credit transactions. Please debit the credit card instead of voiding the credit.
3	180	An error occurred during processing. Please try again.	The processor response format is invalid.
3	181	An error occurred during processing. Please try again.	The system-generated void for the original invalid transaction failed. (The original transaction included an invalid processor response format.)
3	185	This reason code is reserved or not applicable to this API.	
4	193	The transaction is currently under review.	The transaction was placed under review by the risk management system.
2	200	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The credit card number is invalid.
2	201	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The expiration date is invalid.
2	202	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The transaction type is invalid.

66

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
2	203	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The value submitted in the amount field is invalid.
2	204	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The department code is invalid.
2	205	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The value submitted in the merchant number field is invalid.
2	206	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	207	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant account is closed.
2	208	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	209	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Communication with the processor could not be established.
2	210	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant type is incorrect.
2	211	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The cardholder is not on file.
2	212	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The bank configuration is not on file
2	213	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant assessment code is incorrect.
2	214	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
			function is currently unavailable.
2	215	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The encrypted PIN field format is invalid.
2	216	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ATM term ID is invalid.
2	217	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced a general message format problem.
2	218	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The PIN block format or PIN availability value is invalid.
2	219	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ETC void is unmatched.
2	220	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The primary CPU is not available.
2	221	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The SE number is invalid.
2	222	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Duplicate auth request (from INAS).
2	223	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced an unspecified error.
2	224	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Please re-enter the transaction.
3	243	Recurring billing is not allowed for this eCheck.Net type.	The combination of values submitted for x_recurring_billing and x_echeck_type is not allowed.

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
3	244	This eCheck.Net type is not allowed for this Bank Account Type.	The combination of values submitted for x_bank_acct_type and x_echeck_type is not allowed.
3	245	This eCheck.Net type is not allowed when using the payment gateway hosted payment form.	The value submitted for x_echeck_type is not allowed when using the payment gateway hosted payment form.
3	246	This eCheck.Net type is not allowed.	The merchant's payment gateway account is not enabled to submit the eCheck.Net type.
3	247	This eCheck.Net type is not allowed.	The combination of values submitted for x_type and x_echeck_type is not allowed.
2	250	This transaction has been declined.	This transaction was submitted from a blocked IP address.
2	251	This transaction has been declined.	The transaction was declined as a result of triggering a Fraud Detection Suite filter.
4	252	Your order has been received. Thank you for your business!	The transaction was accepted, but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
4	253	Your order has been received. Thank you for your business!	The transaction was accepted and was authorized, but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
2	254	Your transaction has been declined.	The transaction was declined after manual review.
3	261	An error occurred during processing. Please try again.	The transaction experienced an error during sensitive data encryption and was not processed. Please try again.
3	270	The line item [item number] is invalid.	A value submitted in x_line_item for the item referenced is invalid.
3	271	The number of line items submitted is not allowed. A maximum of 30	The number of line items submitted exceeds the allowed

Last revised: 8/30/2010

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		line items can be submitted.	maximum of 30.
3	288	Merchant is not registered as a Cardholder Authentication participant. This transaction cannot be accepted.	The merchant has not indicated participation in any Cardholder Authentication Programs in the Merchant Interface.
3	289	This processor does not accept zero dollar authorization for this card type.	Your credit card processing service does not yet accept zero dollar authorizations for Visa credit cards. You can find your credit card processor listed on your merchant profile.
3	290	One or more required AVS values for zero dollar authorization were not submitted.	When submitting authorization requests for Visa, the address and zip code fields must be entered.
3	300	The device ID is invalid.	The value submitted for x_device_id is invalid.
3	301	The device batch ID is invalid.	The value submitted for x_device_batch_id is invalid.
3	302	The reversal flag is invalid.	The value submitted for x_reversal is invalid.
3	303	The device batch is full. Please close the batch.	The current device batch must be closed manually from the POS device.
3	304	The original transaction is in a closed batch.	The original transaction has been settled and cannot be reversed.
3	305	The merchant is configured for auto-close.	This merchant is configured for auto-close and cannot manually close batches.
3	306	The batch is already closed.	The batch is already closed.
1	307	The reversal was processed successfully.	The reversal was processed successfully.
1	308	Original transaction for reversal not found.	The transaction submitted for reversal was not found.
3	309	The device has been disabled.	The device has been disabled.
1	310	This transaction has already been voided.	This transaction has already been voided.
1	311	This transaction has already been	This transaction has already been

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
		captured	captured.
3	312	The specified security code was invalid.	The customer entered the wrong security code. A new security code will be generated, and the customer will be prompted to try again until successful.
3	313	The customer requested a new security xode.	The customer requested a new security code. A new security code will be generated, and the customer will be prompted to try again until successful.
2	315	The credit card number is invalid.	This is a processor-issued decline.
2	316	The credit card expiration date is invalid.	This is a processor-issued decline.
2	317	The credit card has expired.	This is a processor-issued decline.
2	318	A duplicate transaction has been submitted.	This is a processor-issued decline.
2	319	The transaction cannot be found.	This is a processor-issued decline.

Note: A very helpful tool for troubleshooting errors is available in our Developer Center at
        http://developer.authorize.net/tools/responsereasoncode.

Last revised: 8/30/2010

Section 7
Test Transactions

You will need to test the payment gateway integration carefully before going live to ensure successful and smooth transaction processing.

Ideally, an integration is tested in the following phases:

First, use an Authorize.Net developer test account to submit test transactions through your integration. In this environment, test transactions are posted to **https://test.authorize.net/gateway/transact.dll**. Although this is a staging environment, its behavior mimics the live payment gateway. Transactions submitted to the test environment using a developer test account are **not** submitted to financial institutions for authorization and are not permanently stored in the Merchant Interface, although they will appear in the Unsettled Transactions for the test account immediately after you submit them.

In order to use this environment, you must have an Authorize.Net developer test account with an associated API Login ID and Transaction Key. Test transactions to this environment are accepted with these credentials only. If you do not have a developer test account, you can sign up for one at http://developer.authorize.net.

Note: You do not need to use Test Mode when testing with a developer test account. For more information about Test Mode, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Once the integration is successfully tested in the developer test environment, the merchant's Authorize.Net Payment Gateway API Login ID and Transaction Key can be plugged into the integration for testing against the live environment. (Developer test account credentials will not be accepted by the live payment gateway.) In this phase, testing can be done in one of two ways:

By including the *x_test_request* field with a value of "TRUE" in the HTML Form POST to **https://secure.authorize.net/gateway/transact.dll**. See the sample below.

Sample 16. Submitting the test request field

```
<INPUT TYPE="HIDDEN" NAME="x_test_request"
VALUE="TRUE">
```

By placing the merchant's payment gateway account in Test Mode in the Merchant Interface. New payment gateway accounts are placed in Test Mode by default. For more information about Test Mode, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/. When processing test transactions in Test Mode, the payment gateway will return a transaction ID of "0." This means you cannot test follow-on transactions, for example, credits, voids, etc., while in Test Mode. To test follow-on transactions, you can either submit "x_test_request=TRUE" as indicated above, or process a test transaction with any valid credit card number in live mode, as explained below.

Last revised: 8/30/2010

> **Note:** **Transactions posted against live merchant accounts using either of the above testing methods are not submitted to financial institutions for authorization and are not stored in the Merchant Interface.**

If testing in the live environment is successful, you are ready to submit live transactions and verify that they are being submitted successfully. Either remove the *x_test_request* field from the HTML Form Post or set it to "FALSE;" or if you are using Test Mode, turn it off in the Merchant Interface. To receive a true response, you must submit a transaction using a real credit card number. You can use any valid credit card number to submit a test transaction. You will be able to void successful transactions immediately to prevent live test transactions from being processed. This can be done quickly on the Unsettled Transactions page of the Merchant Interface. It is recommended that when testing using a live credit card, you use a nominal value, such as $0.01. That way, if you forget to void the transaction, the impact will be minimal. For VISA verification transactions, you can submit a $0.00 value instead, if the credit card processor accepts it.

> **Note**: VISA verification transactions are being switched from $0.01 to $0.00 for all processors. For Visa transactions using $0.00, the Bill To address (x_address) and zip code (x_zip) fields are required.

Testing to Generate Specific Transaction Results

When testing transaction results in the developer test environment as well as the production environment, you can produce a specific response reason code by submitting a test transaction using a test credit card number designed to generate specific transaction results: Visa test credit card number "4222222222222." This card number is intended for testing and should only be used for that purpose. Submit the test transaction by either placing the account in Test Mode, or submitting x_test_request=TRUE, with a dollar amount value equal to the response reason code you would like to produce.

For example, to test the AVS response reason code number 27, submit the test transaction with the credit card number "4222222222222" and the amount "27.00."

To test the AVS or CCV responses in the live environment, you will need to submit live transactions with correct street address, ZIP Code and Card Code information to generate successful responses, and incorrect street address, ZIP Code and Card Code information to generate other responses. You can void successful transactions immediately to prevent live test transactions from being processed. This can be done quickly on the Unsettled Transactions page of the Merchant Interface. It is not possible to test the AVS or CCV responses in the developer test environment. For more information about AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

For more information about response reason codes, see the "Transaction Response" section of this guide.

Appendix A
Fields by Transaction Type

This appendix provides a complete listing of all API fields that should be submitted for each transaction type supported for SIM. It is divided into the following sections:

- the minimum fields required to submit a transaction,
- additional fields that are required in order to configure advanced features of SIM, and
- "best practice" fields, or fields that the payment gateway recommends should be submitted on a per-transaction basis in order to maintain a strong connection to the payment gateway—for example, to prevent possible conflicts in the event that integration settings in the Merchant Interface are inadvertently changed.

Minimum Required Fields

The following table provides a quick reference of all API fields that are required for each transaction type supported for SIM.

	AUTHORIZATION AND CAPTURE	AUTHORIZATION ONLY	PRIOR AUTHORIZATION AND CAPTURE*	CREDIT *	VOID*
Merchant Information	x_login	x_login	N/A	N/A	N/A
Fingerprint Information	x_fp_hash x_fp_sequence x_fp_timestamp	x_fp_hash x_fp_sequence x_fp_timestamp	N/A	N/A	N/A
Transaction Information	x_type = AUTH_CAPTURE	x_type = AUTH_ONLY	N/A	N/A	N/A
Payment Information	x_amount	x_amount	N/A	N/A	N/A
Payment Form Configuration	x_show_form = PAYMENT_FORM	x_show_form = PAYMENT_FORM	N/A	N/A	N/A

Last revised: 8/30/2010

*For Prior Authorization and Capture, Credit, and Void transactions, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Required Fields for Additional SIM Features

The following table provides a quick reference of additional fields that are required for advanced features of SIM and that *cannot* be configured in the Merchant Interface. For example, if the merchant wants to submit itemized order information, you must submit fields in addition to the minimum required fields.

	AUTHORIZATION AND CAPTURE	AUTHORIZATION ONLY	PRIOR AUTHORIZATION AND CAPTURE*	CREDIT*	VOID*
Itemized Order Information	x_line_item	x_line_item	N/A	N/A	N/A
Relay Response Configuration	x_relay_response = TRUE x_relay_url	x_relay_response = TRUE x_relay_url	N/A	N/A	N/A
Fraud Detection Suite™ (FDS)	x_customer_ip (required only when the merchant is using the FDS IP blocking tool)	x_customer_ip (required only when the merchant is using the FDS IP blocking tool)	N/A	N/A	N/A

*For Prior Authorization and Capture, Credit, and Void transactions, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Best Practice Fields

The following table provides a quick reference of additional API fields that the payment gateway highly recommends should be submitted on a per-transaction basis in order to maintain a strong connection.

	AUTHORIZATION AND CAPTURE	AUTHORIZATION ONLY	PRIOR AUTHORIZATION AND CAPTURE*	CREDIT*	VOID*
Transaction Information	x_version = 3.1	x_version = 3.1	x_version = 3.1	x_version = 3.1	x_version = 3.1
Payment Form Configuration	x_header_html_payment_form x_footer_html_paymen	x_header_html_payment_form x_footer_html_paymen	N/A	N/A	N/A

Last revised: 8/30/2010

	t_form	t_form			
Receipt Page Configuration	x_receipt_link_method x_header_html_receipt x_footer_html_receipt	x_receipt_link_method x_header_html_receipt x_footer_html_receipt	N/A	N/A	N/A

*For Prior Authorization and Capture, Credit, and Void transactions, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.

Appendix B
Alphabetized List of API Fields

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
x_address	Optional	The customer's billing address	Up to 60 characters (no symbols)	Required if the merchant would like to use the Address Verification Service security feature. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/. Required for Zero Dollar Authorizations for Visa verification transactions.
x_amount	Required if x_type = AUTH_CAPTURE, AUTH_ONLY, CREDIT	The amount of the transaction	Up to 15 digits with a decimal point (no dollar symbol) Ex. 8.95	The total amount to be charged or credited *including* tax, shipping and any other charges. The amount can either be hard coded or posted to a script.
x_background_url	Optional	The URL of the merchant's background image		The image referenced by this URL is displayed as the background on the hosted payment form or receipt page. Background images must be uploaded to the payment gateway server. See the "Logos and background images for the payment form" section of this guide for more information on how to upload images.
x_card_num	Required only if x_type =	The customer's partial credit	The last four digits of the credit card	As a hosted solution, SIM does not support the submission of full cardholder data. Ideally,

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
	CREDIT	card number	number *only*	CREDIT transactions should be submitted exclusively in the Merchant Interface. This field should not be passed for a SIM transaction under any other circumstance.
x_city	Optional	The city of the customer's billing address	Up to 40 characters (no symbols)	
x_color_background d	Optional	The hosted payment form or receipt page background color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the background color for both.
x_color_link	Optional	The hosted payment form and receipt page hyperlink color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the color of the HTML links for both.
x_color_text	Optional	The hosted payment form and receipt page text color	Any valid HTML color name or color hex code	This field is common to the hosted payment form and receipt page. The value in this field will set the color of the text for both.
x_company	Optional	The company associated with the customer's billing address	Up to 50 characters (no symbols)	
x_country	Optional	The country of the customer's billing address	Up to 60 characters (no symbols)	
x_cust_id	Optional	The merchant assigned	Up to 20 characters (no	The unique identifier to represent the customer

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		customer ID	symbols)	associated with the transaction.
				The customer ID must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.
				Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
x_customer_ip	Optional	The customer's IP address	Up to 15 characters (no letters) Ex. 255.255.255.25 5	The IP address of the customer initiating the transaction. If this value is not passed, it will default to 255.255.255.255. This field is required when using the Fraud Detection Suite™ (FDS) IP Address Blocking tool. For more information about FDS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_description	Optional	The transaction description	Up to 255 (no symbols)	The description must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
x_delim_data	Required for SIM transactions	Set to False to implement a relay response.	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	In order to implement a relay response, this field must be submitted with a value of FALSE or the merchant has to configure a relay response through the Merchant Interface. It is recommended that you

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
				submit this field on a per-transaction basis to be sure that transaction responses are returned in the correct format.

This field is paired with x_relay_response. If one is set to True, the other must be set to False. |
| x_duplicate_window | Optional | The window of time after the submission of a transaction that a duplicate transaction can not be submitted | Any value between 0 and 28800 (no commas) | Indicates in seconds the window of time after a transaction is submitted during which the payment gateway will check for a duplicate transaction. The maximum time allowed is 8 hours (28800 seconds).

If a value less than 0 is sent, the payment gateway will default to 0 seconds. If a value greater than 28800 is sent, the payment gateway will default to 28800. If no value is sent, the payment gateway will default to 2 minutes (120 seconds).

If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See the "Response for Duplicate Transactions" section of this guide for more information. |
| x_duty | Optional | The valid duty amount OR delimited duty information | When submitting delimited duty information, values must be delimited by a bracketed pipe <\|> | The duty amount charged OR when submitting this information by means of the HTML Form POST, delimited duty information including the duty name, description, and amount is also allowed.

The total amount of the transaction in x_amount must *include* this amount. |
| | | duty item | | The duty item name. |

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		name<|>		
		duty description<|>		The duty item description.
		duty amount	The dollar sign ($) is not allowed when submitting delimited information.	The duty amount. The total amount of the transaction in x_amount must *include* this amount.
	Example:	`<INPUT TYPE="HIDDEN" name="x_duty" VALUE="Duty1<|>export <|>15.00>`		
x_email	Optional	The customer's valid email address	Up to 255 characters Ex. janedoe@customer.com	The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid. For more information about Email Receipts, please see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_email_customer	Optional	The customer email receipt status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates whether an email receipt should be sent to the customer. If set to TRUE, the payment gateway will send an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer. If no value is submitted, the payment gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has enabled the setting. If this field is not submitted and the setting

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
				is disabled in the Merchant Interface, no email is sent.
				For more information about configuring Email Receipts in the Merchant Interface, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
x_fax	Optional	The fax number associated with the customer's billing address	Up to 25 digits (no letters) Ex. (123)123-1234.	
x_first_name	Optional	The first name associated with the customer's billing address	Up to 50 characters (no symbols)	
x_footer_email_receipt	Optional	The email receipt footer	Plain text	This text will appear as the footer on the email receipt sent to the customer.
x_footer_html_payment _form	Optional	The hosted payment form footer	Plain text or HTML Avoid using double quotes.	The text or HTML submitted in this field is displayed as the footer on the hosted payment form. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with the HTML Form POST. With this method there is no character limit.
x_footer_html_receipt	Optional	The hosted receipt page footer	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed at the bottom of the hosted receipt page. When using HTML styles or referencing a cascading style

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
				sheet (.css), it is recommended that you submit this field with the HTML Form POST. With this method there is no character limit.
x_fp_hash	Required	The transaction unique fingerprint	N/A	The fingerprint is generated using the HMAC-MD5 hashing algorithm on the following field values: API Login ID (x_login) The sequence number of the transaction (x_fp_sequence) The timestamp of the sequence number creation (x_fp_timestamp) Amount (x_amount) Field values are concatenated and separated by the "^" character.
x_fp_sequence	Required	The merchant assigned sequence number for the transaction	Numeric	The sequence number can be a merchant assigned value, such as an invoice number or any randomly generated number.
x_fp_timestamp	Required	The timestamp at the time of fingerprint generation	UTC time in seconds since January 1, 1970	Coordinated Universal Time (UTC) is an international atomic standard of time (sometimes referred to as GMT). Using a local time zone timestamp will cause fingerprint authentication to fail.
x_freight	Optional	The valid freight amount OR delimited freight information	When submitting delimited freight information, values must be delimited by a bracketed pipe	The freight amount charged OR when submitting this information by means of the HTML Form POST, delimited freight information including the freight name, description, and amount is also allowed. The total amount of the

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
			<|>	transaction in x_amount must *include* this amount.
		freight item name<|>		The freight item name.
		freight description<|>		The freight item description.
		freight amount	The dollar sign ($) is not allowed when submitting delimited information.	The freight item amount. The total amount of the transaction in x_amount must *include* this amount.
		Example:	`<INPUT TYPE="HIDDEN" name="x_freight" VALUE="Freight1<|> ground overnight<|>12.95>`	
x_header_email_receipt	Optional	The email receipt header	Plain text	This text will appear as the header of the email receipt sent to the customer.
x_header_html_payment_form	Optional	The hosted payment form header	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed as the header on the hosted payment form. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with the HTML Form POST. With this method there is no character limit.
x_header_html_receipt	Optional	The hosted receipt page header	Plain text or HTML Avoid using double quotes	The text or HTML submitted in this field is displayed at the top of the hosted receipt page. When using HTML styles or referencing a cascading style sheet (.css), it is recommended that you submit this field with the HTML Form POST. With this method there is no character limit.
x_invoice_num	Optional	The merchant	Up to 20	The invoice number must be

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		assigned invoice number for the transaction	characters (no symbols)	created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be included on the hosted payment form, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
x_last_name	Optional	The last name associated with the customer's billing address	Up to 50 characters (no symbols)	
x_line_item	Optional All line item values are required when this field is submitted	Any string	Line item values must be delimited by a bracketed pipe <|>	Itemized order information.
		Item ID<|>	Up to 31 characters	The ID assigned to an item.
		<|>item name<|>	Up to 31 characters	A short description of an item.
		<|>item description<|>	Up to 255 characters	A detailed description of an item.
		<|>itemX quantity<|>	Up to two decimal places Must be a positive number	The quantity of an item.
		<|>item price (unit cost)<|>	Up to two decimal places Must be a positive number	Cost of an item per unit, *excluding* tax, freight, and duty. The dollar sign ($) is not allowed when submitting delimited information.
		<|>itemX	TRUE, FALSE,	Indicates whether the item is

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		taxable	T, F, YES, NO, Y, N, 1, 0	subject to tax.
x_login	Required	The merchant's unique API Login ID	Up to 20 characters	The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and transaction fingerprint together provide the merchant authentication required for access to the payment gateway. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for more information.
x_logo_url	Optional	The URL of the merchant's logo		The image referenced by this URL is displayed in the header or footer of the hosted payment form and receipt page. Logo images must be uploaded to the payment gateway server. See the "Logos and background images for the payment form" section of this guide for more information on how to upload images.
x_method	Optional	The payment method	CC or ECHECK	The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If left blank, this value defaults to CC. For more information about eCheck.Net transaction requirements, see the *eCheck.Net Developer Guide* at http://developer.authorize.net/guides/echeck.pdf.
x_phone	Optional	The phone number associated	Up to 25 digits (no letters)	

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		with the customer's billing address	Ex. (123)123-1234	
x_po_num	Optional	The merchant assigned purchase order number	Up to 25 characters (no symbols)	The purchase order number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function. Also, in order to be displayed, the attribute **View** must be configured for this field in the Merchant Interface payment form settings.
x_receipt_link_method	Optional	The type of link back to the merchant's website from the hosted receipt page	LINK, POST, or GET	LINK creates a regular hyperlink. GET creates a button and returns transaction information in the receipt link URL. POST creates a button and returns transaction information as an HTML Form POST.
x_receipt_link_text	Optional	The text of the link or button that directs the customer back to the merchant's website	Up to 50 characters	If the x_receipt_link_method is LINK, the value in this field will become a hyperlinked text on the hosted receipt page If the x_receipt_link_method is GET or POST the value in this field becomes the text of a submit button. An HTML form is created in the receipt page that has hidden fields containing the results of the transaction processed.
x_receipt_link_url	Optional	The URL of the link or button that directs the customer back to the merchant's		To be accepted as valid by the payment gateway, the URL must be configured in the Merchant Interface. If the x_receipt_link_method is LINK, the URL specified becomes the href value of the

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
		website		hyperlinked text. If the x_receipt_link_method is GET or POST, the URL will become the action of the HTML form.
x_recurring_billing	Optional	The recurring billing status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicating marker used by merchant account providers to identify transactions which originate from merchant hosted recurring billing applications. This value is not affiliated with Automated Recurring Billing.
x_relay_response	Optional	The request for a relay response	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	This field instructs the payment gateway to return transaction results to the merchant by means of an HTML form POST to the merchant's Web server for a relay response. This field is paired with x_delim_data. If one is set to True, the other must be set to False.
x_relay_URL	Optional	The URL on the merchant's website to which the payment gateway should post transaction results for a relay response	Any valid URL Including name/value pairs in the URL(anything after a "?") is not recommended	In the event that this field is submitted, the payment gateway will validate the URL value against the Relay Response URL configured in the Merchant Interface. If the URL submitted does not match the URL configured in the Merchant Interface, the transaction will be rejected. If no value is submitted in the HTML Form POST, the payment gateway will post the transaction results to the URL configured in the Merchant Interface.
x_rename	Optional	A request to rename a field	Field name on the payment form, new field name	Use this variable to replace a field name on a payment form. This does not rename the original field, it only changes the value displayed on the

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
				payment form. See section Renaming a Field for more information.
x_ship_to_address	Optional	The customer's shipping address	Up to 60 characters (no symbols)	
x_ship_to_company	Optional	The company associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_country	Optional	The country of the customer's shipping address	Up to 60 characters (no symbols)	
x_ship_to_city	Optional	The city of the customer's shipping address	Up to 40 characters (no symbols)	
x_ship_to_first_name	Optional	The first name associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_last_name	Optional	The last name associated with the customer's shipping address	Up to 50 characters (no symbols)	
x_ship_to_state	Optional	The state of the customer's shipping address	Up to 40 characters (no symbols) or a valid two-character state	

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
			code	
x_ship_to_zip	Optional	The ZIP code of the customer's shipping address	Up to 20 characters (no symbols)	
x_show_form	Required	The payment form request	PAYMENT_FORM	The show form field indicates that the merchant wishes to use the payment gateway hosted payment form to collect payment data.
x_state	Optional	The state of the customer's billing address	Up to 40 characters (no symbols) or a valid two-character state code	
x_tax	Optional	The valid tax amount OR the delimited tax information	When submitting delimited tax information, values must be delimited by a bracketed pipe <|>	The tax amount charged OR when submitting this information by means of the HTML Form POST, delimited tax information including the sales tax name, description, and amount is also allowed. The total amount of the transaction in x_amount must *include* this amount.
		tax item name<|>		The tax item name.
		tax description<|>		The tax item description.
		tax amount	The dollar sign ($) is not allowed when submitting delimited information.	The tax item amount. The total amount of the transaction in x_amount must *include* this amount.
	Example:	`<INPUT TYPE="HIDDEN" name="x_tax" VALUE="Tax1<|>state tax<|>0.0625">`		

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
x_tax_exempt	Optional	The tax exempt status	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates whether the transaction is tax exempt. The total amount of the transaction in x_amount must *include* this amount.
x_test_request	Optional	The request to process test transactions	TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0	Indicates if the transaction should be processed as a test transaction. See the "Test Transactions" section of this guide for more information.
x_trans_id	Optional	The payment gateway-assigned transaction ID of an original transaction	Numeric	Required only for CREDIT, PRIOR_AUTH_CAPTURE, and VOID transactions. For more information about transaction types, see the "Credit Card Transaction Types" section of this guide.
x_type	Optional	The type of credit card transaction	AUTH_CAPTURE (default), AUTH_ONLY	If the value submitted does not match a supported value, the transaction is rejected. If no value is submitted in this field, the payment gateway will process the transaction as an AUTH_CAPTURE. For transaction types CREDIT, PRIOR_AUTH_CAPTURE, and VOID, it is recommended that the merchant process the transactions by logging on to the Merchant Interface directly, or by using a desktop application that uses AIM.
x_version	Optional, but highly recommended	The merchant's transaction version	3.1	The transaction version represents the set of fields that is included with the transaction response. 3.0 is the default version. 3.1 allows the merchant to utilize the Card Code feature, and is the current standard version. It is highly recommended that

Last revised: 8/30/2010

FIELD NAME	REQUIRED?	VALUE	FORMAT	NOTES
				you submit this field on a per-transaction basis to be sure that the formats of transaction requests and the responses you receive are consistent, particularly if you are using Relay Response. For more information, see the "Relay Response" and "Appendix A Fields by Transaction Type" sections of this document.
x_zip	Optional	The ZIP code of the customer's billing address	Up to 20 characters (no symbols)	Required if the merchant would like to use the Address Verification Service security feature. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/. Required for Zero Dollar Authorizations for Visa verification transactions.

Index

SIM Developer Guide