

Card-Present Transactions Implementation Guide

Authorize.Net LLC (“Authorize.Net”) has made efforts to ensure the accuracy and completeness of the information in this document. However, Authorize.Net disclaims all representations, warranties and conditions, whether express or implied, arising by statute, operation of law, usage of trade, course of dealing or otherwise, with respect to the information contained herein. Authorize.Net assumes no liability to any party for any loss or damage, whether direct, indirect, incidental, consequential, special or exemplary, with respect to (a) the information; and/or (b) the evaluation, application or use of any product or service described herein.

Authorize.Net disclaims any and all representation that its products or services do not infringe upon any existing or future intellectual property rights. Authorize.Net owns and retains all right, title and interest in and to the Authorize.Net intellectual property, including without limitation, its patents, marks, copyrights and technology associated with the Authorize.Net services. No title or ownership of any of the foregoing is granted or otherwise transferred hereunder. Authorize.Net reserves the right to make changes to any information herein without further notice.

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®



Table of Contents

| | |
|--|-----------|
| REVISION HISTORY | 4 |
| INTRODUCTION | 5 |
| Connection Methods..... | 5 |
| Advanced Integration Method (AIM) | 5 |
| Customer and Developer Support | 6 |
| SECTION 2 INTEGRATION SETTINGS | 8 |
| Features of the Gateway | 8 |
| Address Verification System | 8 |
| Credit Card Identification Code (CVV2/CVC2/CID) | 9 |
| What is the Advanced Integration Method (AIM)? | 11 |
| How Does AIM Work?..... | 11 |
| What is Required to Implement AIM? | 11 |
| The AIM Application Program Interface (API) | 11 |
| Software Development Kits (SDK) | 12 |
| SUBMITTING TRANSACTIONS..... | 13 |
| Minimum Requirements..... | 13 |
| Credit Card Transaction Processing..... | 14 |
| Credit Card Transaction Types | 14 |
| Partial Authorization Transactions | 16 |
| STANDARD CP TRANSACTION SUBMISSION API FOR AIM..... | 18 |
| Merchant Account Information..... | 18 |
| Gateway Response Configuration | 20 |
| Customer Name and Billing Address..... | 20 |
| Additional Customer Data..... | 21 |
| Invoice Information | 22 |
| Itemized Order Information..... | 22 |
| Customer Shipping Address | 23 |
| Transaction Data | 24 |
| Track1 and Track2 Data | 24 |
| Level 2 Data | 28 |

| | |
|--|-----------|
| GATEWAY RESPONSE API | 29 |
| Delimited Response | 29 |
| Fields in the Gateway Response | 29 |
| Response for Duplicate Transactions | 31 |
| XML Response | 31 |
| Response Code Details | 34 |
| Description of Response Fields | 34 |
| Response Codes | 34 |
| Response Reason Codes & Response Reason Text..... | 35 |
| HTTP Error Codes & Reason Text..... | 46 |
| APPENDIX A – SUBMITTING TEST TRANSACTIONS TO THE SYSTEM | 47 |
| Test Mode..... | 47 |
| Running a Test Transaction | 47 |
| Testing to Generate Specific Transaction Results | 48 |
| APPENDIX B – RESPONSE EXAMPLES | 49 |
| Sample Delimited Responses..... | 49 |
| Sample XML Responses | 49 |
| APPENDIX C – THE TRANSACTION KEY | 52 |
| What is the Transaction Key?..... | 52 |
| How do I Obtain the Transaction Key? | 52 |
| APPENDIX D – THE SECRET QUESTION AND ANSWER..... | 53 |
| What is the Secret Question and Answer? | 53 |
| How do I Set my Secret Question and Answer?..... | 53 |
| APPENDIX E – TRACK DATA | 54 |
| Track 1 Data..... | 54 |
| Track 2 Data | 56 |
| APPENDIX F – CURRENCY CODES | 57 |

Revision History

| PUBLISH DATE | UPDATES |
|---------------|---|
| | |
| June 2009 | Additions to the Response Code list |
| June 2010 | Partial Authorization implementation Format of guide updated |
| October 2010 | Corrected table of response fields to allow for up to 26 fields |
| November 2010 | Added reference to enabling Transaction Details API |
| May 2011 | Format update |

Introduction

Welcome to the Authorize.Net Card Present (CP) Implementation Guide. This guide describes how card present transactions can be submitted to the gateway for real-time processing.

Connection Methods

You can submit transactions to Authorize.Net in the following ways:

- Automatically through a website or custom application connected to Authorize.Net using Advanced Integration Method (AIM).
- Ready-to-go point-of-sale (POS) solutions include our own Virtual Point of Sale as well as numerous third party card swipe terminals, software, self-service kiosks, and mobile devices. A list of our partners' [Certified POS Solutions](#) can be found in our [Certified Solution Directory](#).

It's a good idea to identify how your business plans to submit transactions so that you and/or your Web developer can properly integrate your payment gateway account to support your business processes.

By communicating your transaction processing practices or requirements, you can help your Web developer integrate your website or custom application more quickly.

Advanced Integration Method (AIM)

AIM is a customizable payment processing solution that gives the merchant control over all the steps in processing a transaction, including:

- Collecting customer payment information through a custom application
- Generating a receipt to the customer
- Secure transmission to the payment gateway for transaction processing
- Secure storage of cardholder information
- And more, depending on the merchant's business requirements

The security of an AIM transaction is assured through a 128-bit Secure Sockets Layer (SSL) connection between the merchant's Web server and the Authorize.Net Payment Gateway.

Payment gateways act as a bridge between a merchant's point-of-purchase software and the financial institutions that process payment transactions. Payment data is submitted by the point-of-purchase software using the Internet to the gateway for real-time authorization.

Authorization is the process of checking the validity and available balance of a customer's credit card before a transaction can be accepted. To authorize a given credit card transaction, the gateway transmits the payment information to the appropriate financial institution for validation, then returns the response (approved or declined) from the institution to the merchant's software.

The Payment Gateway supports real-time and offline requests for credit card authorization.

Note: For CP transactions, the merchant and the shopper are in the same physical location. The merchant has a card reader (or "swipe terminal") and receipt printer at the point of purchase. The card reader device reads the magnetic stripe on the back of the card and transmits the encoded information to the gateway. Once a transaction is approved, the merchant can print a receipt for obtaining the cardholder's signature.

Customer and Developer Support

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

- Refer to the Merchant Interface Online Help Files. Log on to the Merchant Interface at <https://account.authorize.net>, click on the feature for which you need help from the main menu or **Settings** menu, and then click the **Help** link in the top right corner of the page.
- The Authorize.Net Knowledge Base, located at <http://www.authorize.net/help>, provides comprehensive answers to virtually any customer support question, as well as useful links to demos, help files and information on contacting us. We strongly recommend using the Knowledge Base anytime you need help.
- Software Development Kits (SDKs) are available in several popular languages from our developer website, <http://developer.authorize.net/downloads/>
- Customer Support is available to help you with questions regarding integration settings in the Merchant Interface. You can contact Customer Support by emailing support@authorize.net, or using chat by clicking **Live Help** in the top right corner of the Merchant Interface. Customer Support hours are 5:00 AM – 5:00 PM Pacific time, Monday through Friday.
- The Developer Center at <http://developer.authorize.net> provides Web developers with test accounts, sample code, FAQs, and troubleshooting tools.
- If you or your developer can't find what you need in the Developer Center, our Developer support team is available to answer your questions by email at developer@authorize.net. (Please note that our Developer support team can only

assist with support requests specific to the Authorize.Net application programming interface (API) and/or services.)

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

Section 2

Integration Settings

Most integration settings in the Merchant Interface apply to both Server Integration Method (SIM) and Advanced Integration Method (AIM). However, some are specific to the connection method you are using. This section details all the settings you should be aware of in the Merchant Interface that will help you achieve and maintain a strong connection to the payment gateway.

Features of the Gateway

The gateway supports the following features in an effort to reduce the merchant's chargeback liability.

Address Verification System

Bankcard processors implemented the Address Verification Service (AVS) to aid merchants in the detection of suspicious transaction activity. The payment processing network compares the billing address provided in the transaction with the cardholder's address on file at the credit card issuing bank. The processing network returns an AVS response code that indicates the results of this comparison to the payment gateway. You can configure your account to reject certain transactions based on the AVS code returned. For example, the AVS code "A" indicates that the street address matched, but the first five digits of the ZIP Code did not.

Note: In most cases, AVS is not required for a card-present transaction since the cardholder is present at the point of purchase. However, in cases where the magnetic stripe reader cannot read the track data, some processors specify that AVS data be sent with the manually keyed-in card information.

The following result codes are possible.

| AVS CODE | DESCRIPTION |
|----------|--|
| 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 | 9 digit ZIP matches, Address (Street) does not |
| X | Address (Street) and 9 digit ZIP match |
| Y | Address (Street) and 5 digit ZIP match |
| Z | 5 digit ZIP matches, Address (Street) does not |

To configure transaction rejection settings based on the AVS response code:

Log on to the Merchant Interface at <https://account.authorize.net>

1. Select **Settings** under Account in the main menu on the left
2. Click **Address Verification Service** in the Security Settings section
3. Click to select the check box(es) next to the AVS codes for which the payment gateway should reject transactions
4. Click **Submit**

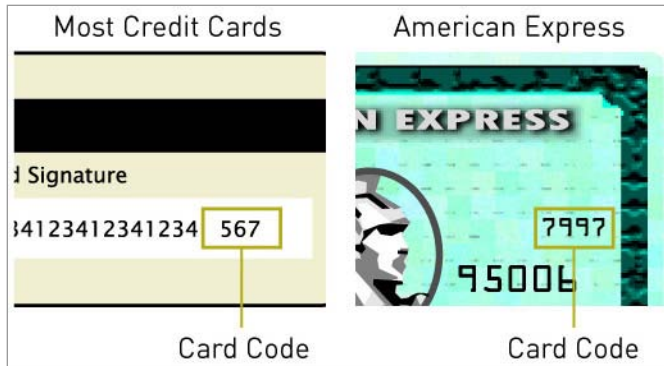
Transactions are processed against your rejection criteria immediately.

Note: In order to use the AVS filter, you need to require the billing address and ZIP code fields when collecting payment information from your customers. Please communicate these requirements to your Web developer.

Credit Card Identification Code (CVV2/CVC2/CID)

The Credit Card Identification Code, or “Card Code,” is a three- or four-digit security code that is printed on the back of credit cards in reverse italics in the card's signature panel (or on the front for American Express cards). The merchant can collect this information from the customer and submit the data to the gateway. The gateway will pass this information to the financial institution along with the credit card number. The financial institution will determine if the value matches the value on file for that credit card and return a code indicating whether the comparison failed or succeeded, in addition to whether the card was authorized. The gateway passes back this response code to the merchant.

Figure 1. Finding the card code on a credit card



You can choose to collect this information from the customer and submit the data to the payment gateway, as another method for authenticating credit card transactions submitted through your account. The payment gateway will pass this information to the credit card issuer along with the credit card number. The credit card issuer will determine if the value matches the value on file for the customer's credit card and return a code to the payment gateway indicating whether the code matched, in addition to indicating whether the card was authorized. You can configure the payment gateway to reject transactions based on the code returned.

| CARD CODE RESPONSE | DESCRIPTION |
|--------------------|---|
| M | Card Code matched |
| N | Card Code does not match |
| P | Card Code was not processed |
| S | Card Code should be on card but was not indicated |
| U | Issuer was not certified for Card Code |

Note: In most cases, Card Codes are not required for a card-present transaction since the cardholder is present at the point of purchase. However, in cases where the magnetic stripe reader cannot read the track data, some processors specify that Card Code data be sent with the manually keyed-in card information.

What is the Advanced Integration Method (AIM)?

Software that resides on a merchant's POS or other IP terminal can submit transactions to the gateway using Advanced Integration Method (AIM). AIM allows a merchant's server to automatically and securely connect to the Payment Gateway to submit transaction data. This method requires merchants to be able to initiate and manage secure Internet connections.

How Does AIM Work?

When using AIM, transactions flow in the following way:

1. The Merchant's server initiates a secure connection to the Payment Gateway and then initiates an HTTPS POST of the transaction data to the gateway server
2. The Payment Gateway receives and processes the transaction data
3. The Payment Gateway then generates and submits the transaction response to the Merchant's server
4. The Merchant's server receives and processes the response
5. Finally, the Merchant prints a receipt and obtains the cardholder's signature to complete the transaction

What is Required to Implement AIM?

Merchants must be able to perform the following functions in order to submit transactions to the gateway using AIM:

- Have a secure socket layer (SSL) digital certificate
- Provide both server and client side encryption
- Develop scripts on a Web server for the integration to the gateway (for example, for submitting transaction data and receiving system responses)

The AIM Application Program Interface (API)

A defined API is provided for submitting transactions to the Payment Gateway. An API is also provided for responses to transactions that are submitted to the gateway. These APIs are discussed in detail in this document.

The merchant uses the Merchant Interface to configure the transaction response from the gateway. (The Merchant Interface is a tool through which merchants can manage their accounts and their transaction activity. A user login ID and password are required to access this tool. The URL to the Merchant Interface is available to the merchant from their merchant service provider.)

Software Development Kits (SDK)

An SDK is available to help you set up an application in PHP, Ruby, Java, and C#. You can download this SDK from <http://developer.authorize.net/api/aim/>.

Submitting Transactions

The payment gateway supports several credit card transaction types for transactions submitted by AIM.

To implement AIM for a merchant's business, a developer would design a script that can do the following:

- Securely obtain all of the information needed to process a transaction
- Initiate a secure HTTPS form POST from the merchant's server to <https://cardpresent.authorize.net/gateway/transact.dll>
- Receive the response from the gateway and process the response to display the appropriate result to the end user

The transaction submission API defines the information that can be submitted to the gateway for real-time transaction processing. The API consists of a set of fields that are required for each transaction, and a set of fields that are optional.

Minimum Requirements

The following table contains the minimum set of NAME/VALUE pairs that must be submitted to the gateway when using AIM. The data fields are name/value pairs with the syntax of:

`x_name_of_field=value of field&`.

| FIELD NAME | FIELD VALUE | NOTES |
|-------------------|---|--|
| x_cpversion | 1.0 | |
| x_login | API Login ID for the payment gateway account | |
| x_market_type | Your market type | |
| x_device_type | Your device type | |
| x_amount | Amount of purchase inclusive of tax | |
| x_split_tender_id | The payment gateway-assigned ID that links the current authorization request to the original authorization request. | This value applies to partial authorization transactions only, and is returned in the reply message from the original authorization request. |
| x_tran_key | Your transaction key | See Appendix F for more information on the transaction key. |
| x_track1 OR | Track 1 data from credit card | Must be supplied if neither x_track2 nor |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD NAME | FIELD VALUE | NOTES |
|----------------|---------------------------------|---|
| | | x_card_num and x_exp_date data is submitted. |
| x_track2 OR | Track 2 data from credit card | Must be supplied if neither x_track1 nor x_card_num and x_exp_date data is submitted. |
| x_card_num AND | Customer's card number | Must be supplied if neither x_track1 nor x_track2 data is submitted. |
| x_exp_date | Customer's card expiration date | Must be supplied if neither x_track1 nor x_track2 data is submitted. |

Note: For security reasons, use only port 443 for AIM information transfers.

Credit Card Transaction Processing

There are two steps to credit card transaction processing:

- **Authorization** is the process of checking the validity and available balance of a customer's credit card before the transaction is accepted. The transaction submission methods describe the request for authorization.
- **Settlement**, also referred to as *Capture*, is the process by which funds are actually transferred from the customer to the merchant for goods and services sold. Based on the transaction type specified in the authorization request, the gateway initiates the settlement step. As part of the settlement process, the gateway will send a settlement request to the financial institution to request transfer of funds. The gateway does not control the time frame within which funds are actually transferred.

The merchant can specify when the last transaction is picked up for settlement by the gateway. To modify the Transaction Cut-Off Time, do the following (only users with the appropriate permissions can access this setting):

1. Log on to the Merchant Interface
2. Select **Settings**
3. Select **Transaction Cut-Off Time** from the General section
4. Using the drop-down boxes, select the desired cut-off time
5. Click **Submit** to save changes

Credit Card Transaction Types

The following table describes the type of transactions that can be submitted to the gateway and how the gateway will process them.

| TRANSACTION TYPE | DESCRIPTION |
|--------------------|--|
| AUTH_CAPTURE | <p>Transactions of this type are sent for authorization. The transaction is automatically picked up for settlement if approved. This is the default transaction type in the gateway. If no type is indicated when submitting transactions to the gateway, the gateway will assume that the transaction is of the type AUTH_CAPTURE.</p> <p>If the approved amount is less than the requested amount and the transaction qualifies as a partial authorization transaction, see the special conditions described in Error! Not a valid bookmark self-reference..</p> |
| AUTH_ONLY | <p>Transactions of this type are submitted if the merchant wishes to validate the credit card for the amount of the goods sold. If the merchant does not have goods in stock or wishes to review orders before shipping the goods, this transaction type should be submitted. The gateway will send this type of transaction to the financial institution for approval. However this transaction will not be sent for settlement. If the merchant does not act on the transaction within 30 days, the transaction will no longer be available for capture.</p> |
| PRIOR_AUTH_CAPTURE | <p>This transaction is used to request settlement for a transaction that was previously submitted as an AUTH_ONLY. The gateway will accept this transaction and initiate settlement if the following conditions are met:</p> <ul style="list-style-type: none"> • The transaction is submitted with the ID of the original authorization-only transaction which needs to be settled. • The transaction ID is valid and the system has a record of the original authorization-only transaction being submitted. • The original transaction referred to is not already settled or expired or errored. • The amount being requested for settlement in this transaction is less than or equal to the original authorized amount. <p>If no amount is submitted in this transaction, the gateway will initiate settlement for the amount of the originally authorized transaction.</p> <hr/> <p>Note: If extended line item, tax, freight, and/or duty information was submitted with the original transaction, adjusted information can be submitted in the event that the transaction amount changed. If no adjusted line item, tax, freight, and/or duty information is submitted, the information submitted with the original transaction will apply.</p> |
| CREDIT | <p>This transaction is also referred to as a "Refund" and indicates to the gateway that money should flow from the merchant to the customer. The gateway will accept a credit or a refund request if the transaction submitted meets the following conditions:</p> <p>The transaction is submitted with the ID of the original transaction against which the credit is being issued (x_ref_trans_id).</p> <ul style="list-style-type: none"> • The gateway has a record of the original transaction. • The original transaction has been settled. • The sum of the amount submitted in the Credit transaction and all credits submitted against the original transaction is less than the original transaction amount. • The full or last four digits of the credit card number submitted with the credit transaction match the full or last four digits of the credit card number used in |

| | |
|--------------|---|
| | <p>the original transaction.</p> <ul style="list-style-type: none"> The transaction is submitted within 120 days of the settlement date of the original transaction. <p>A transaction key is required to submit a credit to the system (i.e., <i>x_tran_key</i> should have a valid value when a CREDIT transaction is submitted).</p> <p>For details about how to submit CREDIT transactions to the Payment Gateway, please see the Issuing Credits Guide at http://www.authorize.net/files/creditreturnssummary.pdf.</p> |
| CAPTURE_ONLY | <p>This is a request to settle a transaction that was not submitted for authorization through the payment gateway. The gateway will accept this transaction if an authorization code is submitted. <i>x_auth_code</i> is a required field for CAPTURE_ONLY type transactions.</p> |
| VOID | <p>This transaction is an action on a previous transaction and is used to cancel the previous transaction and ensure it does not get sent for settlement. It can be done on any type of transaction (i.e., CREDIT, AUTH_CAPTURE, CAPTURE_ONLY, and AUTH_ONLY). The transaction is accepted by the gateway if the following conditions are met:</p> <ul style="list-style-type: none"> The transaction is submitted with the ID of the transaction that has to be voided. The gateway has a record of the transaction referenced by the ID. The transaction has not been sent for settlement. |

Partial Authorization Transactions

A *split tender order* is one in which two or more transactions are used to cover the total amount of the order. The merchant must indicate that they are able to handle the extra processing either by selecting the Partial Authorization option in the Account settings of the Merchant Interface, or by sending *x_allow_partial_auth=true* with an individual transaction. Without this flag, the transaction would be handled as any other, and would be either fully authorized or declined due to lack of funds on the card.

When the first transaction is successfully approved for a partial amount of the total order, a split tender ID is generated and returned to the merchant in the response. This ID must be passed back with each of the remaining transactions of the group, using *x_split_tender_id=<value>*. If you include both a split tender ID and a transaction ID on the same request, an error results.

If successfully authorized, all transactions in the group are held until the final transaction of the group is successfully authorized, unless the merchant has indicated either by input parameter or default configuration that they do not want these transactions to be held.

If the merchant needs to release the group of transactions before the final one is approved (if the balance is paid by cash, for example), send a `prior_auth_capture` request and include the split tender ID instead of a transaction ID.

If the merchant needs to void the group before completion, a void request should be sent, using the split tender ID instead of a transaction ID.

The following rules apply to partial authorization transactions:

- The merchant can choose to accept partial authorization transactions by selecting an option in the Merchant Interface. Alternatively, partial authorization transactions can be submitted by including a new API field (`x_allow_partial_auth`) in the initial request that enables partial authorization for that specific request...
- When an authorization is granted for an amount less than the purchase amount, a Split Tender ID is provided (`x_split_tender_id`), in addition to the Transaction ID. The Split Tender ID is used on subsequent payments for that purchase, instead of the Transaction ID.
- The transaction is not submitted for settlement until either the merchant submits payments adding up to the full requested amount, or the merchant indicates the transaction has been completed (in the case when all or part of the remaining balance is paid in cash).
- Any of the transactions can be voided using the Split Tender ID.
- The Split Tender ID cannot be submitted together with a Transaction ID; only one or the other can be submitted.

Unique field requirements for Partial Authorization Transactions are:

- `x_allow_partial_auth=TRUE` (input, optional) —The default value is set in the Merchant Interface; you can use this parameter to authorize individual transactions if the option is set to False in the Merchant Interface. Including this field in the transaction request overrides the merchant's account configuration.
- `x_prepaid_balance_on_card` (output)—this is the authorized amount remaining on the card.
- `x_prepaid_requested_amount` (output)—this is the amount requested.
- `x_split_tender_id` (output)—this is the split tender ID provided when the first partial authorization transaction was issued. Use this ID when submitting subsequent transactions related to the same group order.
- `x_split_tender_status` (output)—indicates whether or not the transaction is complete.
- `x_card_type` (output)—the card type.

Standard CP Transaction Submission API for AIM

The Card Present 1.0 API supports several Required, Conditional, and Optional information fields for submitting transaction data to the credit card processors and card associations. Some information fields are supported by the API, but are not required by the payment gateway for submitting basic transactions. However, some of these fields might be required by your acquiring bank to meet their transaction processing requirements. You or your Web developer should contact your acquiring bank to learn about their specific transaction information requirements.

The following tables list the transaction data fields that can be submitted using the transaction request string. Several of these fields 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/>.

Fields are name/value pairs with the syntax:

```
x_name_of_field = value of the field&
```

Merchant Account Information

The following fields in the API allow the system to identify the merchant submitting the transaction and the state of the merchant's account on the gateway.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|---------------|----------|--|------------|---|
| x_login | Required | Varies by Merchant | N/A | Pass the API Login ID for the payment gateway account. |
| x_tran_key | Required | Varies by Merchant System-generated value obtained from the Merchant Interface. | N/A | The transaction key is similar to a password and is used by the system to authenticate requests that are submitted to the gateway. See Appendix F for instructions on how to obtain the transaction key from your Merchant Interface. |
| x_market_type | Required | 2 2 = Retail | N/A | The market type that is configured for your account. |

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|----------------------|----------|--|------------|--|
| x_device_type | Required | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 1 = Unknown 2 = Unattended Terminal 3 = Self Service Terminal 4 = Electronic Cash Register 5 = Personal Computer- Based Terminal 6 = AirPay 7 = Wireless POS 8 = Website 9 = Dial Terminal 10 = Virtual Terminal | N/A | The device type that is configured for your account. |
| x_cpversion | Optional | 1.0 | N/A | Determines the feature set used to process the transaction request. Version 1.0 is currently the only supported version. |
| x_test_request | Optional | TRUE, YES, Y, ON, or 1 (case insensitive) Any of these values indicate a test transaction. Any other value will result in a live transaction. | N/A | When Test Mode is turned on from the Merchant Interface, it cannot be overridden (set to FALSE) using this NAME/VALUE pair. The system default is FALSE. Please refer to Appendix D for more information on Test Mode. |
| x_response_format | Optional | 0, 1 0 = XML 1 = Delimited | N/A | Determines the format of the system response to a transaction request. The system default is "0" or XML. |
| x_user_ref | Optional | Any value supplied by the merchant. | 255 | User reference field provided by the system for the merchant's use. The value of this field will return to the merchant in the response exactly as it was submitted. |
| x_allow_partial_Auth | Optional | True, False | N/A | Indicates if the transaction is enabled for partial authorization. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|-------|----------|-------|------------|---|
| | | | | Including this field in the transaction request overrides your account configuration. |

Gateway Response Configuration

The following fields determine how a transaction response is returned after a transaction is submitted to the system. The merchant has the option of sending in the configuration of the response on a per-transaction basis or configuring the response through the Merchant Interface. Submitting values in these fields on a per-transaction basis overrides the configuration in the Merchant Interface for that transaction. It is recommended that the values be set in the Merchant Interface for these fields and not submitted on a per-transaction basis.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|--------------|----------|---------------------|------------|---|
| x_delim_char | Optional | Any valid character | 1 | Character that is used to separate fields in the transaction response. The system will use the character passed in this field or the value stored in the Merchant Interface if no value is passed. |
| x_encap_char | Optional | Any valid character | 1 | Character that is used to encapsulate the fields in the transaction response. The system will use the character passed in this field or the value stored in the Merchant Interface if no value is passed. |

Customer Name and Billing Address

The customer billing address fields listed below contain information on the customer billing address associated with each transaction.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|--------------|----------|------------|------------|---|
| x_first_name | Optional | Any string | 50 | Contains the first name of the customer associated with the billing address for the |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|-------------|----------|---|------------|--|
| | | | | transaction. |
| x_last_name | Optional | Any string | 50 | Contains the last name of the customer associated with the billing address for the transaction. |
| x_company | Optional | Any string | 50 | Contains the company name associated with the billing address for the transaction. |
| x_address | Optional | Any string | 60 | Contains the address of the customer associated with the billing address for the transaction. |
| x_city | Optional | Any string | 40 | Contains the city of the customer associated with the billing address for the transaction. |
| x_state | Optional | Any string | 40 | Contains the state of the customer associated with the billing address for the transaction. |
| x_zip | Optional | Any string | 20 | Contains the zip of the customer associated with the billing address for the transaction. |
| x_country | Optional | Any string | 60 | Contains the country of the customer associated with the billing address for the transaction. |
| x_phone | Optional | Any string Recommended format is (123)123-1234 | 25 | Contains the phone number of the customer associated with the billing address for the transaction. |
| x_fax | Optional | Any string Recommended format is (123)123-1234 | 25 | Contains the fax number of the customer associated with the billing address for the transaction. |

Additional Customer Data

Merchants can provide additional customer information with a transaction, based on their respective requirements.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|-----------|----------|------------|------------|--|
| x_cust_id | Optional | Any string | 20 | Unique identifier to represent the customer associated with the transaction. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|---------|----------|-------------------------|------------|---|
| x_email | Optional | Any valid email address | 255 | Email address to which the customer's copy of the confirmation email is sent. No email is sent to the customer if the email address does not meet standard email format checks. |

Invoice Information

Based on their respective requirements, merchants can submit invoice information with a transaction. Two invoice fields are provided in the gateway API.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|---------------|----------|------------|------------|-----------------------------------|
| x_invoice_num | Optional | Any string | 20 | Merchant-assigned invoice number. |
| x_description | Optional | Any string | 255 | Description of the transaction. |

Itemized Order Information

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

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|------------------------|--|--|------------|-----------------------------|
| x_line_item | Optional Required if the order is itemized. | Any string. Line item values must be delimited by < >. | N/A | Itemized order information. |
| item ID< > | Optional Required if the order is itemized. | Any string | 31 | Item ID. |
| < >item name< > | Optional Required if the order is itemized. | Any string | 31 | Item name. |
| < >item description< > | Optional Required if the order | Any string | 255 | Item description. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|-------------------------------|---|--|------------|---|
| | is itemized. | | | |
| < >itemX quantity< > | Optional Required if the order is itemized | Any positive number (two decimal places allowed) | N/A | Item quantity. |
| < >item price (unit cost) < > | Optional Required if the order is itemized | Any positive number (two decimal places allowed) | N/A | Item unit price, excluding tax, freight and duty. The dollar sign (\$) is not allowed when submitting delimited information. |
| < >itemX taxable | Optional Required if the order is itemized | YES, NO | N/A | Indicates whether the item is taxable. |

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

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

Note: For Prior_Auth_Capture transactions, if line item information was submitted with the original transaction, adjusted information can be submitted if the transaction changed. If no adjusted line item information is submitted, the information submitted with the original transaction will apply.

Customer Shipping Address

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

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|----------------------|----------|------------|------------|--|
| x_ship_to_first_name | Optional | Any string | 50 | Contains the customer shipping first name. |
| x_ship_to_last_name | Optional | Any string | 50 | Contains the customer shipping last |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|-------------------|----------|------------|------------|---|
| | | | | name. |
| x_ship_to_company | Optional | Any string | 50 | Contains the customer shipping company. |
| x_ship_to_address | Optional | Any string | 60 | Contains the customer shipping address. |
| x_ship_to_city | Optional | Any string | 40 | Contains the customer shipping city. |
| x_ship_to_state | Optional | Any string | 40 | Contains the customer shipping state. |
| x_ship_to_zip | Optional | Any string | 20 | Contains the customer shipping zip. |
| x_ship_to_country | Optional | Any string | 60 | Contains the customer shipping country. |

Transaction Data

The fields in the following table contain the transaction-specific information such as amount, payment method, and the transaction type.

Track1 and Track2 Data

The transaction-specific information can include track 1 and track 2 data from the magnetic strip on the credit card. Although the entire unaltered track must be provided in the authorization request message, any framing characters (Start and End Sentinel) must be removed first:

- For track 1 data, the Start Sentinel character is a percent sign (%); the End Sentinel character is a question mark (?).
- For track 2 data, the Start Sentinel is ASCII \$0B, while the End Sentinel character is ASCII \$0F. If the bytes in track 2 data are converted to ASCII, this turns the Start Sentinel character to a semicolon (;), and the End Sentinel to a question mark (?).

The Start Sentinel and End Sentinel characters must be removed from any data submitted for x_track1 and x_track2.

For more detailed information about Track 1 and Track 2 data, see Appendix F – Track Data on page 54.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|----------|----------|------------|------------|---------------------------------------|
| x_amount | Required | Any amount | 15 | Total value to be charged or credited |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|---------------------|--|--|------------|--|
| | | | | <p>inclusive of tax. Mandatory for all x_type values except for VOID, and PRIOR_AUTH_CAPTURE when the capture amount equals the original amount authorized.</p> <p>The system will remove dollar signs and commas and allows for no more than one decimal point; the remaining characters must be numbers.</p> |
| x_currency_code | Optional | Valid currency code | 3 | Currency of the transaction amount. If left blank, this will default to the value specified in the Merchant Interface. |
| x_method | Optional | CC | N/A | Indicates the method of payment for the transaction being sent to the system. |
| x_type | Required | AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT, VOID, PRIOR_AUTH_CAPTURE | N/A | <p>Indicates the type of transaction. If no value is submitted in this field, the gateway will process the transaction as an AUTH_CAPTURE.</p> <p>If the value in the field does not match any of the values stated, the transaction is rejected.</p> |
| x_recurring_billing | Optional | YES, NO | 3 | Indicates whether the transaction is a recurring billing transaction. |
| x_track1 | Conditional Required only if x_track2, x_card_num, and x_exp_date are absent. | Valid Track 1 data Note: Starting and ending sentinel characters must be discarded before submitting transactions. | N/A | <p>Track 1 data read from credit card. This information is required only if Track 2 data and x_card_num and x_exp_date are absent. It is not necessary to submit Track 1 <i>and</i> Track 2 data <i>and</i> x_card_num and x_exp_date. If both tracks are sent by the POS application, the gateway will use the Track 1 information.</p> <p>If neither Track 1 nor Track 2 data is submitted, but x_card_num and x_exp_date are submitted, the Card Present transaction rate might be downgraded.</p> <p>See Appendix F – Track Data for more information on Track Data formats.</p> |
| x_track2 | Conditional Required only if x_track1 and | Valid Track 2 data Note: Starting and ending sentinel | N/A | Track 2 data read from credit card. This information is required only if Track 1 and x_card_num and x_exp_date are absent. It is not |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|----------------|---|--|------------|---|
| | x_card_num, and x_exp_date are absent. | characters must be discarded before submitting transactions. | | necessary to submit Track 1 <i>and</i> Track 2 data <i>and</i> x_card_num and x_exp_date. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but x_card_num and x_exp_date are submitted, the Card Present transaction rate might be downgraded. See Appendix F – Track Data for more information on Track Data formats. |
| x_card_type | The customer's credit card type | V = Visa M = MasterCard A = American Express D = Discover C = Diners Club J = JCB | | |
| x_card_num | Conditional Required when Track 1 or Track 2 data is absent, or for manually entered/keyed-in transactions and credit transactions (x_type = CREDIT) | Numeric credit card number | 22 | Credit card number. (Provided when track data is absent.) If neither Track 1 nor Track 2 data is submitted, but x_card_num and x_exp_date are submitted, the Card Present transaction rate might be downgraded. See Appendix F – Track Data for more information on Track Data formats. |
| x_exp_date | Conditional Required when Track 1 or Track 2 data is absent, or for manually entered/keyed-in transactions and credit transactions (x_type = CREDIT) | MMYY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY, YYYY-MM-DD, YYYY/MM/DD | 20 | Date on which the credit card expires Note: The system will reject single digit year values (for example, 05/3). If neither Track 1 nor Track 2 data is submitted, but x_card_num and x_exp_date are submitted, the Card Present transaction rate might be downgraded. See Appendix F – Track Data for more information on Track Data formats. |
| x_card_code | Optional | Any valid card code CVV2 (Visa), CVC2 (MasterCard), CID (AMEX) | 4 | The three- or four-digit number on the back of a credit card. If the card code is passed with track data, x_card_code is ignored. |
| x_ref_trans_id | Conditional Required if | Any valid transaction ID | 10 | ID of a transaction previously authorized by the gateway. If passed with other types of transactions, |

Last revised: 5/24/2011

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|--------------------|--|---|------------|--|
| | x_type = CREDIT, VOID or PRIOR_AUTH_CAPTURE | | | x_ref_trans_id is ignored. |
| x_split_tender_id | Conditional | The payment gateway-assitned ID assigned when the original transaction includes two or more partial payments. This is the identifier that is used to group transactions that are part of a split tender order. | | If the first transaction results in a partial authorization, the payment gateway returns this ID to the merchant. The merchant must pass this ID back with each subsequent transaction that is part of the group of transactions sent to obtain the entire amount of the order. The payment gateway does not calculate new amounts, that is up to the merchant's software. For more information about partial authorization transactions, see the "Partial Authorization Transactions" topic. |
| x_auth_code | Conditional Required if x_type = CAPTURE_ONLY | Any valid authorization code | 6 | Authorization code for a previous transaction not authorized on the gateway that is being submitted for Capture. If an authorization code with a transaction not specified as CAPTURE_ONLY is sent, the system will ignore x_auth_code. |
| x_duplicate_window | Optional | Any value between 0 – 28800 | | 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 28000 sent, the payment gateway will default to 28000. 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. Please see the section of this document titled "Response for Duplicate Transactions" for more information. |

Level 2 Data

The system supports Level 2 transaction data by providing the following fields as part of the transaction submission API. The tax, freight, and duty fields allow a delimited string for submitting extended information.

| FIELD | REQUIRED | VALUE | MAX LENGTH | DESCRIPTION |
|--------------|----------|--|------------|--|
| x_po_num | Optional | Any string | 25 | Contains the purchase order number. |
| x_tax | Optional | Any valid tax amount OR the following delimited values: tax item name < > tax description< > tax amount | N/A | Contains the tax amount OR delimited tax information including the sales tax name, description, and amount. The dollar sign (\$) is not allowed when submitting delimited information. |
| x_tax_exempt | Optional | TRUE, FALSE | 5 | Indicates whether the transaction is tax exempt. |
| x_freight | Optional | Any valid freight amount OR the following delimited values: freight item name < > freight description< > freight amount | N/A | Contains the freight amount charged OR delimited freight information including the freight name, description, and amount. The dollar sign (\$) is not allowed when submitting extended information. |
| x_duty | Optional | Any valid duty amount OR the following delimited values: duty item name < > duty description< > duty amount | N/A | Contains the amount charged for duty OR delimited duty information including the duty name, description, and amount. The dollar sign (\$) is not allowed when submitting extended information. |

Note: For Prior_Auth_Capture transactions, if extended tax, freight, and/or duty information was submitted with the original transaction, adjusted information can be submitted in the event that the transaction amount changed. If no adjusted tax, freight, and/or duty information is submitted, the information submitted with the original transaction will apply.

Gateway Response API

The transaction response from the payment gateway is returned as a delimited string and provides information about the status of a transaction—whether it was accepted or declined—as well as information included in the transaction request.

Delimited Response

This section describes the response returned by the gateway when a merchant server submits a transaction for processing. The response is a set of fields that give the merchant information about the status of a transaction. The fields are bar ('|') delimited by default, or delimited by the character specified by the merchant. The merchant server can parse this data and then determine the message to display to the customer. The delimited response uses mime type text/plain. When `x_response_format` is set to "1," the system will return a bar ('|') delimited text response according to the table below.

Fields in the response are delimited by a character that is specified in the transaction request string (`x_delim_char`) or configured in the Merchant Interface. The merchant server can parse this data to customize receipt messages to display or email to the customer. Transaction results are also provided in the payment gateway merchant confirmation email, and on the Transaction Detail page for the transaction in the Merchant Interface.

Fields in the Gateway Response

The following table indicates the order of the fields returned in the response from the gateway to the merchant server.

| POSITION IN RESPONSE | FIELD RETURNED | DESCRIPTION |
|----------------------|--------------------|--|
| 1 | Version | System version used to process the transaction. |
| 2 | Response Code | Indicates the result of the transaction: 1 = Approved 2 = Declined 3 = Error 4 = Held for Review |
| 3 | Reason Code | A code representing more details about the result of the transaction. |
| 4 | Reason Text | Brief description of a result, which corresponds with the Reason Code. |
| 5 | Authorization Code | Contains the six-digit alphanumeric approval code. |

| POSITION IN RESPONSE | FIELD RETURNED | DESCRIPTION |
|----------------------|--------------------|--|
| 6 | AVS Code | Indicates the result of Address Verification System (AVS) checks: 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 = 9 digit ZIP matches, Address (Street) does not X = Address (Street) and 9 digit ZIP match Y = Address (Street) and 5 digit ZIP match Z = 5 digit ZIP matches, Address (Street) does not |
| 7 | Card Code Response | Indicates the results of Card Code verification: M = Match N = No Match P = Not Processed S = Should have been present U = Issuer unable to process request |
| 8 | Transaction ID | This number identifies the transaction in the system and can be used to submit a modification of this transaction at a later time, such as voiding, crediting or capturing the transaction. |
| 9 | MD5 Hash | System-generated hash that can be validated by the merchant to authenticate a transaction response received from the gateway. |
| 10 | User Reference | Echoed by the system from the form input field <code>x_user_ref</code> . |
| 11-20 | reserved | These fields are reserved |
| 21 | Card number | The card number is returned as XXXX1234 |
| 22 | card type | Visa, MasterCard, American Express, Discover, Diners Club, JCB |
| 23 | Split tender ID | The value that links the current authorization request to the original authorization request. This value is returned in the reply message from the original authorization request. This is only returned in the reply message for the first transaction that receives a partial authorization. |
| 24 | Requested amount | The amount requested in the original authorization |
| 25 | Approved amount | The amount approved for this transaction |
| 26 | Remaining balance | Balance on the debit card or prepaid card. This has a value only if the |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| POSITION IN RESPONSE | FIELD RETURNED | DESCRIPTION |
|----------------------|----------------|---|
| | on card | current transaction is for a prepaid card |

Note: Fields 23 through 26 are only provided if the merchant has enabled partial authorization transactions.

Response for Duplicate Transactions

The Card Present API allows you to specify the window of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction. To use this functionality, you must pass the Duplicate Window (*x_duplicate_window*) field with a value between 0 to 28800 seconds (maximum of 8 hours).

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

If 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 gateway response for the duplicate transaction will also include 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 Code result
- The Card Code result
- The Transaction ID
- The MD5 Hash
- The User Reference

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 default 2 minute duplicate window, are processed normally.

XML Response

When *x_response_format* is set to "0," or no value is supplied, the response will return XML as follows:

```

<?xml version="1.0" ?>
<response>
  <ResponseCode>xx</ResponseCode>
  <Errors> (0 or 1)
    <Error> (1 or more)
      <ErrorCode>xx</ErrorCode>
      <ErrorText>
        <![CDATA[xxxxxx]]
      </ErrorText>
    </Error>
  </Errors>
  <Messages> (0 or 1)
    <Message> (1 or more)
      <Code>xx</Code>
      <Description>
        <![CDATA[xxxxxx]]
      </Description>
    </Message>
  </Messages>
  <AuthCode>
    <![CDATA[xxxxxxx]]
  </AuthCode>
  <AVSResultCode>xx</AVSResultCode>
  <CVVResultCode>xx</CVVResultCode>
  <TransID>xxxxxx</TransID>
  <RefTransID>xxxxxxx</RefTransID>
  <TransHash>xxxxxxx</TransHash>
  <TestMode>0|1</TestMode>
  <UserRef>xxxxxx</UserRef>
</response>

```

| FIELD RETURNED | DESCRIPTION |
|----------------|--|
| Response code | Indicates the result of the transaction: 1 = Approved 2 = Declined 3 = Error |
| Errors | There can be 0 or 1 occurrences of this tag. If the response code is approved then there will be 0 occurrence of this tag. |

| FIELD RETURNED | DESCRIPTION |
|----------------|--|
| Error | There can be 1 or more occurrence if the errors tag exists. |
| Error code | Indicates the type of error. |
| Error text | Description of the error. |
| Messages | There can be 0 or 1 occurrence of this tag. If the response code is declined or error then there will be 0 occurrence of this tag. |
| Message | There can be 1 or more occurrence if the messages tag exists. |
| Code | Indicates the type of message |
| Description | Description of the message |
| Auth code | 6 digit alphanumeric authorization code. |
| AVSResultCode | Indicates the result of Address Verification System (AVS) checks: 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 = 9 digit ZIP matches, Address (Street) does not X = Address (Street) and 9 digit ZIP match Y = Address (Street) and 5 digit ZIP match Z = 5 digit ZIP matches, Address (Street) does not |
| CVVResultCode | Indicates the results of Card Code verification: M = Match N = No Match P = Not Processed S = Should have been present U = Issuer unable to process request |
| Transid | This number identifies the transaction in the system and can be used to submit a modification of this transaction at a later time, such as voiding, crediting or capturing the transaction. |
| Reftransid | Value passed in or 0 if it doesn't apply and wasn't passed in. |
| Transhash | System-generated hash that can be validated by the merchant to authenticate a transaction response received from the gateway. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| FIELD RETURNED | DESCRIPTION |
|----------------|--|
| Testmode | Value of 0 indicates the transaction was in live mode |
| Userref | Echoed by the system from the form input field x_user_ref. |

Response Code Details

When a payment transaction is submitted to the gateway, the gateway returns a response that indicates the general status of the transaction, including details of what caused the transaction to be in that state. The fields in the response that describe the status of the transaction are: Response Code, Response Reason Code, and Response Reason Text. The following tables define the values that the gateway might return in these fields.

Description of Response Fields

The three status fields in the transaction response are defined as follows:

- The **Response Code** indicates the overall status of the transaction with possible values of approval, decline, or error.
- The **Response Reason Code** gives merchants more information about the transaction status.
- The **Response Reason Text** is a text string that will give more detail on why the transaction resulted in a specific response code. This field is a text string that can be echoed back to the customer to provide them with more information about their transaction. It is strongly suggested that merchants not parse this string expecting certain text. Instead, a merchant should test for the Response Reason Code if they need to programmatically know these results; the Response Reason Code will always represent these meanings, even if the text descriptions change.

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 & 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. | The card was declined by the issuer. Contact the card issuer to determine the reason. |
| 2 | 3 | This transaction has been declined. | This code indicates a referral response. Contact the card issuer to complete the transaction. |
| 2 | 4 | This transaction has been declined. | The card was reported lost or stolen. Contact the card issuer for resolution. |
| 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. | |
| 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 | This reason code is reserved or not applicable to this API. | |
| 3 | 10 | This reason code is reserved or not applicable to this API. | |
| 3 | 11 | A duplicate transaction has been submitted. | A transaction with identical amount and credit card information was submitted two minutes prior. |
| 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 | This reason code is reserved or not applicable to this API. | |
| 3 | 15 | The transaction ID is invalid. | The transaction ID value is non-numeric or was not present for a transaction that requires it (i.e., VOID, PRIOR_AUTH_CAPTURE, and CREDIT). |
| 3 | 16 | The transaction was not found. | The transaction ID sent in was properly formatted but the gateway had no record of |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|---|--|
| | | | the transaction. |
| 3 | 17 | The merchant does not accept this type of credit card. | The merchant was not configured to accept the credit card submitted in the transaction. |
| 3 | 18 | This reason code is reserved or not applicable to this API. | |
| 3 | 19 | An error occurred during processing. Please try again in 5 minutes. | Processor error - Could not connect to processor for authorization. If these errors persist, contact merchant service provider. |
| 3 | 20 | An error occurred during processing. Please try again in 5 minutes. | Processor error - Timed out while waiting for a response from processor for authorization. If these errors persist, contact the merchant service provider. |
| 3 | 21 | An error occurred during processing. Please try again in 5 minutes. | A database error occurred during processing. |
| 3 | 22 | An error occurred during processing. Please try again in 5 minutes. | A general error occurred during processing. |
| 3 | 23 | An error occurred during processing. Please try again in 5 minutes. | Processor error - An error occurred during authorization. If these errors persist, contact merchant service provider. |
| 3 | 24 | This reason code is reserved or not applicable to this API. | |
| 3 | 25 | An error occurred during processing. Please try again in 5 minutes. | Processor error - An error occurred during authorization. If these errors persist, contact merchant service provider. |
| 3 | 26 | An error occurred during processing. Please try again in 5 minutes. | Processor error – the processor is not available. If these errors persist, contact the merchant service provider. |
| 2 | 27 | The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder. | Authorization with the card issuer was successful but the transaction was declined due to an address or ZIP code mismatch with the address on file with the card issuing bank based on the settings in the Merchant Interface. |
| 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. Contact the merchant service provider for resolution. |
| 2 | 29 | This reason code is reserved or not applicable to this API. | |
| 2 | 30 | The configuration with the processor is invalid. Call | The merchant account is not configured properly with the Credit Card Processor. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|--|--|
| | | Merchant Service Provider. | Contact the merchant service provider for resolution. |
| 2 | 31 | This reason code is reserved or not applicable to this API. | |
| 3 | 32 | This reason code is reserved or not applicable to this API. | |
| 3 | 33 | <i>FIELD</i> cannot be left blank. | The word <i>FIELD</i> is replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in. |
| 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 Provider. | The merchant was incorrectly set up at the processor. |
| 3 | 36 | The authorization was approved, but settlement failed. | |
| 2 | 37 | The credit card number is invalid. | Processor error - Invalid Credit Card Number. Call merchant service provider for resolution. |
| 2 | 38 | This reason code is reserved or not applicable to this API. | |
| 3 | 39 | The supplied currency code is either invalid, not supported, not allowed for this merchant or doesn't have an exchange rate. | |
| 3 | 40 | This transaction must be encrypted. | |
| 2 | 41 | This reason code is reserved or not applicable to this API. | |
| 3 | 42 | This reason code is reserved or not applicable to this API. | |
| 3 | 43 | This reason code is reserved or not applicable to this API. | |
| 2 | 44 | This reason code is reserved or not applicable to this API. | |
| 2 | 45 | This reason code is reserved or not applicable to this API. | |
| 3 | 46 | This reason code is reserved or | |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|--|--|
| | | not applicable to this API. | |
| 3 | 47 | The amount requested for settlement must 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 \$[amount] will not be accepted. | The transaction amount submitted was greater than the maximum 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 | This reason code is reserved or not applicable to this API. | |
| 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 reason code is reserved or not applicable to this API. | |
| 3 | 57 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 58 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 59 | An error occurred in processing. Please try again in 5 minutes. | |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|----------------------|-----------------------------|---|--|
| 3 | 60 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 61 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 62 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 63 | An error occurred in processing. Please try again in 5 minutes. | |
| 3 | 64 | This reason code is reserved or not applicable to this API. | |
| 2 | 65 | This reason code is reserved or not applicable to this API. | |
| 3 | 66 | This transaction cannot be accepted for processing. | The transaction did not meet gateway security guidelines. |
| 3 | 67 | This reason code is reserved or not applicable to this API. | |
| 3 | 68 | The version parameter is invalid. | The value submitted in x_cpversion was invalid. |
| 3 | 69 | The transaction type is invalid. | The value submitted in x_type was invalid. |
| 3 | 70 | The transaction method is invalid. | The value submitted in x_method was invalid. |
| 3 | 71 | This reason code is reserved or not applicable to this API. | |
| 3 | 72 | The authorization code is invalid. | The value submitted in x_auth_code was more than six characters in length. |
| 3 | 73 | This reason code is reserved or not applicable to this API. | |
| 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 | This reason code is reserved or | |

Last revised: 5/24/2011

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|---|--|
| | | not applicable to this API. | |
| 3 | 78 | The Card Code (CVV2/CVC2/CID) is invalid. | The value submitted in x_card_code failed format validation. |
| 3 | 79 | This reason code is reserved or not applicable to this API. | |
| 3 | 80 | This reason code is reserved or not applicable to this API. | |
| 3 | 81 | This reason code is reserved or not applicable to this API. | |
| 3 | 82 | This reason code is reserved or not applicable to this API. | |
| 3 | 83 | This reason code is reserved or not applicable to this API. | |
| 3 | 84 | The device type is invalid. | The value submitted in x_device_type did not match the configured value. |
| 3 | 85 | The market type is invalid. | The value submitted in x_market_type did not match the configured value. |
| 3 | 86 | The response format is invalid. | The value submitted in x_response_format was not equal to "0" or "1." |
| 3 | 87 | This market type is not supported. | |
| 3 | 88 | The Track1 data is invalid. | |
| 3 | 89 | The Track2 data is invalid. | |
| 3 | 90 | ACH transactions cannot be processed. | ACH transactions cannot be processed by the card-present system. |
| 3 | 91 | This reason code is reserved or not applicable to this API. | |
| 3 | 92 | This reason code is reserved or not applicable to this API. | |
| 3 | 93 | This reason code is reserved or not applicable to this API. | |
| 3 | 94 | This reason code is reserved or not applicable to this API. | |
| 3 | 95 | This reason code is reserved or not applicable to this API. | |
| 3 | 96 | This reason code is reserved or | |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|---|---|
| | | not applicable to this API. | |
| 3 | 97 | This reason code is reserved or not applicable to this API. | |
| 3 | 98 | This reason code is reserved or not applicable to this API. | |
| 3 | 99 | This reason code is reserved or not applicable to this API. | |
| 3 | 100 | This reason code is reserved or not applicable to this API. | |
| 3 | 101 | This reason code is reserved or not applicable to this API. | |
| 3 | 102 | This reason code is reserved or not applicable to this API. | |
| 3 | 103 | This transaction cannot be accepted. | A valid fingerprint, transaction key, or password is required for this transaction. |
| 3 | 104 | This reason code is reserved or not applicable to this API. | |
| 3 | 105 | This reason code is reserved or not applicable to this API. | |
| 3 | 106 | This reason code is reserved or not applicable to this API. | |
| 3 | 107 | This reason code is reserved or not applicable to this API. | |
| 3 | 108 | This reason code is reserved or not applicable to this API. | |
| 3 | 109 | This reason code is reserved or not applicable to this API. | |
| 3 | 110 | This reason code is reserved or not applicable to this API. | |
| 3 | 111 | A valid billing country is required. | This code is applicable to Wells Fargo SecureSource SM merchants only. |
| 3 | 112 | A valid billing state/province is required. | This code is applicable to Wells Fargo SecureSource SM merchants only. |
| 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.) |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|---|---|
| 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 processing. Please try again. | The system-generated void for the 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. |
| 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. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|----------------------|-----------------------------|---|--|
| 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 transaction cannot be processed. | Merchant is not configured for VPOS. |
| 4 | 193 | This reason code is reserved or not applicable to this API. | |
| 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. |
| 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 | This error code applies only to merchants on |

Last revised: 5/24/2011

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|----------------------|-----------------------------|-------------------------------------|--|
| | | declined. | 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 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 | This error code applies only to merchants on |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|---|--|
| | | declined. | FDC Omaha. Please re-enter the transaction. |
| 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 line items can be submitted. | The number of line items submitted in x_line_item exceeds the allowed maximum of 30. |
| 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. |
| 4 | 295 | The amount of this request was only partially approved on the given prepaid card. Additional payments are required to complete the balance of this transaction. | |
| 3 | 296 | The specified SplitTenderId is not valid. | |
| 3 | 297 | A Transaction ID and a Split Tender ID cannot both be used in a single transaction request. | |
| 3 | 300 | The device ID is invalid. | The value submitted for x_device_id is not valid. |
| 3 | 301 | The device batch ID is invalid. | The value submitted for x_device_batch_id is not valid. |
| 3 | 302 | The reversal flag is invalid. | The value submitted for x_reversal is not valid. |
| 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 | The reversal was processed successfully. |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| RESPONSE CODE | RESPONSE REASON CODE | RESPONSE REASON TEXT | NOTES |
|---------------|----------------------|--|---|
| | | 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 captured | This transaction has already been captured. |
| 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: Response code reasons that are not included in numerical order are reserved, or might not be applicable to this API.

HTTP Error Codes & Reason Text

| HTTP CODE | RESPONSE REASON TEXT | NOTES |
|-----------|--|---|
| 503 | Our servers are currently too busy to handle your request. Please wait a minute and resubmit. Thank you. | The payment gateway has momentarily reached transaction queuing capacity. |

Appendix A – Submitting Test Transactions to the System

Test Mode

Test Mode is a special mode of interacting with the system that is useful during the initial setup phase, allowing merchants to test their setup without processing live card data.

To set an account to Test Mode, do the following (only users with the appropriate permissions are able to access this setting):

1. Log on to the Merchant Interface
6. Select **Settings** from the Main Menu
7. Click the **Test Mode** Link in the General section
8. Click the Turn Test On button

In Test Mode, all transactions appear to be processed as real transactions. The gateway accepts the transactions, but does not pass them on to the financial institutions. Accordingly, all transactions are approved by the gateway when Test Mode is turned on. Transactions submitted in Test Mode are not stored on the system, and do not appear in any reports or lists.

Running a Test Transaction

It is possible to run a test transaction if Test Mode has been turned off. This can be done by indicating to the gateway in the transaction submission request that the transaction should be processed as a test transaction. The corresponding field in the transaction submission API is *x_test_request*. If a test transaction is desired, the value of this field should be set to TRUE.

The following table describes the gateway behavior based on the incoming field value and the mode configured through the Merchant Interface.

| VALUE PASSED IN X_TEST_REQUEST | CONFIGURATION IN MERCHANT INTERFACE | GATEWAY BEHAVIOR |
|--------------------------------|-------------------------------------|-------------------------------|
| TRUE | ON | Transaction processed as test |
| FALSE | ON | Transaction processed as test |
| TRUE | OFF | Transaction processed as test |

| VALUE PASSED IN X_TEST_REQUEST | CONFIGURATION IN MERCHANT INTERFACE | GATEWAY BEHAVIOR |
|--------------------------------|-------------------------------------|---|
| FALSE | OFF | Transaction processed as live transaction |

If there is no value submitted in the *x_test_request* field, the system will use the configuration specified in the Merchant Interface.

Testing to Generate Specific Transaction Results

To cause the system to generate a specific error, set the account to Test Mode and submit a transaction with the card number 422222222222. The system will return the response reason code equal to the amount of the submitted transaction. For example, to test response reason code number 27, a test transaction would be submitted with the credit card number, "422222222222," and the amount, "27.00."

Appendix B – Response Examples

Sample Delimited Responses

Approval Response

```
1.0|1|1|This transaction has been
approved.|ABC123|A|M|1002313|FFEEDDCCBBAA99887766554433221100|ID1234
```

Decline Response

```
1.0|2|4|This transaction has been
declined.|000000|||0|FFEEDDCCBBAA99887766554433221100|ID1234
```

Error Response

```
1.0|3|53|Invalid credit card
number.|000000|||0|FFEEDDCCBBAA99887766554433221100|ID1234
```

Sample XML Responses

Approval Response

```
<?xml version="1.0" ?>
<response>
  <ResponseCode>1</ResponseCode>
  <Messages>
    <Message>
      <Code>1</Code>
      <Description><![CDATA[This transaction has been approved.]]></Description>
    </Message>
  </Messages>
  <AuthCode><![CDATA[ABCD]]></AuthCode>
  <AVSResultCode>P</AVSResultCode>
  <CVVResultCode></CVVResultCode>
  <TransID>106707002</TransID>
  <RefTransID>0</RefTransID>
  <TransHash>BC46B890B5495B0FB419DE97CB5DAE9C</TransHash>
  <TestMode>0</TestMode>
</response>
```

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

```
<UserRef></UserRef>
</response>
```

Decline Response

```
<?xml version="1.0" ?>
<response>
  <ResponseCode>2</ResponseCode>
  <Errors>
    <Error>
      <ErrorCode>2</ErrorCode>
      <ErrorText><![CDATA[This transaction has been declined.]]></ErrorText>
    </Error>
  </Errors>
  <AuthCode><![CDATA[]]></AuthCode>
  <AVSResultCode>P</AVSResultCode>
  <CVVResultCode></CVVResultCode>
  <TransID>106707003</TransID>
  <RefTransID>0</RefTransID>
  <TransHash>4852F60CD7D22CB31E98397E6F20673E</TransHash>
  <TestMode>0</TestMode>
  <UserRef></UserRef>
</response>
```

Error Response

```
<?xml version="1.0" ?>
<response>
  <ResponseCode>3</ResponseCode>
  <Errors>
    <Error>
      <ErrorCode>33</ErrorCode>
      <ErrorText><![CDATA[Credit card number is required.]]></ErrorText>
    </Error>
    <Error>
      <ErrorCode>5</ErrorCode>
      <ErrorText><![CDATA[A valid amount is required.]]></ErrorText>
    </Error>
  </Errors>
  <AuthCode><![CDATA[]]></AuthCode>
  <AVSResultCode>P</AVSResultCode>
  <CVVResultCode></CVVResultCode>
```

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

```
<TransID>0</TransID>
<RefTransID>0</RefTransID>
<TransHash>B663878ED0F52E88168B30DBACE92D47</TransHash>
<TestMode>0</TestMode>
<UserRef></UserRef>
</response>
```

Note: *<ErrorCode>* contains the Response Reason Code in the case of declines and errors, as documented in the Implementation Guides. Also, in accordance with XML standards, the element and attribute names and values are case sensitive.

Appendix C – The Transaction Key

What is the Transaction Key?

The gateway-generated transaction key is similar to a password and is used by the system to authenticate requests that are submitted to the gateway. (The transaction key is submitted with a transaction using the *x_tran_key* form field.) Merchants can obtain a unique transaction key through the Settings menu of the Merchant Interface.

How do I Obtain the Transaction Key?

To obtain the transaction key, follow these steps (only users with the appropriate permissions are able to access this setting):

1. Log on to the Merchant Interface
9. Select **Settings** from the main menu
10. Click **Obtain Transaction Key** in the Security section
11. Enter your **Secret Answer** (see Appendix G)
12. You can select **Disable Old Transaction Key**. If you do not select this box, the old transaction key automatically expires in one day.
13. Click **Submit**. The new transaction key is displayed.
14. Copy and paste your new transaction key to a safe location. Once it is displayed, the transaction key will not appear again (although a new one can be generated).

Appendix D – The Secret Question and Answer

What is the Secret Question and Answer?

The secret question and answer are used to protect vital account information and settings. Merchants use their secret question and answer when obtaining the transaction key (See Appendix F) or when contacting Customer Support for assistance.

Upon first login, all new merchants are required to specify a secret question and answer. Merchants should protect this secret question and answer at all times and only disclose them to individuals with privileged access to sensitive account information.

How do I Set my Secret Question and Answer?

Merchants can update the secret question and answer at any time through the Merchant Interface. When choosing a secret question and answer, be sure to select a question and answer that cannot be guessed easily. (For example: if your pet's name is a commonly used name, the question, "What is your pet's name" would not make an ideal candidate for a secret question.)

To change the secret question and answer, follow these steps (only users with the appropriate permissions are able to access this setting):

1. Log on to the Merchant Interface
2. Select **Settings** from the main menu
3. Click the **Change Secret Question/Answer** link in the Security section
4. Enter your **Current Secret Answer**
5. Select a new Secret Question
6. Enter the new Secret Answer
7. Click **Submit** to save changes

Appendix E – Track Data

Accurate Track 1 or Track 2 data is required to receive Card Present rates. Authorization requests containing altered Track 1 or Track 2 data are flagged as NOT COMPLIANT by Visa and MasterCard, resulting in the merchant paying the highest transaction rate and forfeiture of chargeback protection. Both associations monitor non-compliant transactions and will assess fines and penalties to merchants that are not in compliance.

The POS device or software must perform the following operations on Track read data before it can be used in an authorization request message.

- The longitudinal redundancy checks (LRC) must be calculated for the data read from the Track and compared to the LRC read from the Track. The Track data is assumed to be read without errors when no character parity errors are detected and the calculated and read LRCs match.
- The starting sentinel, ending sentinel, and LRC are discarded.
- The character codes read from the magnetic stripe must be converted from the encoded character set to the set used for the authorization request message. The characters encoded on Track 1 are six bit plus parity codes and the characters encoded on Track 2 are four bit plus parity codes, with the character set used for the request message defined as seven bit plus parity code. All characters read from a Track must be converted to the request message character set and transmitted as part of the request. the converted Track data can not be modified by adding or deleting non-framing characters and must be a one for one representation of the characters read from the Track.

Note: You only need to submit Track 1 *or* Track 2 data. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but *x_card_num* and *x_exp_date* are submitted, the Card Present transaction rate might be downgraded.

Track 1 Data

This is a variable length field with a maximum data length of 76 characters.

The Track 1 data read from the cardholder's card is checked for parity and LRC errors, and then converted from the six-bit characters encoded on the card to seven bit characters as defined in ANSI X3.4.

As part of the conversion, the terminal must remove the framing characters (start sentinel, end sentinel, and LRC characters). The separators must be converted to either

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

an ASCII “^” (HEX 5E) or ASCII <US> (HEX 1F) characters. The entire unaltered track, excluding framing characters, must be provided in the authorization request message, or an error condition will result.

Track 1 can be encoded with up to 79 characters as shown below:

| | | | | | | | | | | |
|----|----|-----|----|------|----|------|-----------|-----------------------|----|-----|
| SS | FC | PAN | FS | NAME | FS | DATE | SVC CD | DISCRETIONARY DATA | ES | LRC |
|----|----|-----|----|------|----|------|-----------|-----------------------|----|-----|

LEGEND:

| FIELD | DESCRIPTION | LENGTH | FORMAT |
|--------------------|-------------------------------------|----------|--------|
| SS | Start Sentinel | 1 | % |
| FC | Format Code(“B” for credit cards) | 1 | A/N |
| PAN | Primary Account Number | 19 max | NUM |
| FS | Field Separator | 1 | ^ |
| FS | | | |
| NAME | Card Holder Name | 2-25 max | A/N |
| FS | Field Separator | 1 | ^ |
| DATE | Expiration Date(YMMM) | 4 | NUM |
| SVC CD | Service Code | 3 | NUM |
| Discretionary Data | Optional Issuer Data | Variable | A/N |
| ES | End Sentinel | 1 | ? |
| LRC | Longitudinal Redundancy Check | 1 | |
| | Total CANNOT exceed 79 bytes— —> | 79 | |

Track 2 Data

This is a variable length field with a maximum data length of 37 characters.

The Track 2 data read from the cardholder's card is checked for parity and LRC errors, and then converted from the four-bit characters encoded on the card to seven bit characters as defined in ANSI X3.4. As part of the conversion, the terminal must remove the start sentinel, end sentinel, and LRC characters. The separators must be converted to either an ASCII "=" (HEX 3D) or ASCII "D" (HEX 44) characters. The entire unaltered track (excluding framing characters) must be provided in the authorization request message, or an error message is generated.

Track 2 Data can be encoded with up to 40 characters as shown below:

| | | | | | | | |
|----|-----|----|------|-----------|-----------------------|----|-----|
| SS | PAN | FS | DATE | SVC CD | DISCRETIONARY DATA | ES | LRC |
|----|-----|----|------|-----------|-----------------------|----|-----|

LEGEND:

| FIELD | DESCRIPTION | LENGTH | FORMAT |
|--------------------|-------------------------------|----------|--------|
| SS | Start Sentinel | 1 | ; |
| PAN | Primary Account Number | 19 max | NUM |
| FS | Field Separator | 1 | = |
| DATE | Expiration Date(YMMM) | 4 | NUM |
| SVC CD | Service Code | 3 | NUM |
| Discretionary Data | Optional Issuer Data | Variable | A/N |
| ES | End Sentinel | 1 | 0F Hex |
| LRC | Longitudinal Redundancy Check | 1 | |
| | Total CANNOT exceed 40 bytes | 40 | |

Appendix F – Currency Codes

| CURRENCY COUNTRY | CURRENCY CODE |
|--|----------------------|
| Afghani (Afghanistan) | AFA |
| Algerian Dinar (Algeria) | DZD |
| Andorran Peseta (Andorra) | ADP |
| Argentine Peso (Argentina) | ARS |
| Armenian Dram (Armenia) | AMD |
| Aruban Guilder (Aruba) | AWG |
| Australian Dollar (Australia) | AUD |
| Azerbaijani Manat (Azerbaijan) | AZM |
| Bahamian Dollar (Bahamas) | BSD |
| Bahraini Dinar (Bahrain) | BHD |
| Baht (Thailand) | THB |
| Balboa (Panama) | PAB |
| Barbados Dollar (Barbados) | BBD |
| Belarussian Ruble (Belarus) | BYB |
| Belgian Franc (Belgium) | BEF |
| Belize Dollar (Belize) | BZD |
| Bermudian Dollar (Bermuda) | BMD |
| Bolivar (Venezuela) | VEB |
| Boliviano (Bolivia) | BOB |
| Brazilian Real (Brazil) | BRL |
| Brunei Dollar (Brunei Darussalam) | BND |
| Bulgarian Lev (Bulgaria) | BGN |
| Burundi Franc (Burundi) | BIF |
| Canadian Dollar (Canada) | CAD |
| Cape Verde Escudo (Cape Verde) | CVE |
| Cayman Islands Dollar (Cayman Islands) | KYD |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| CURRENCY COUNTRY | CURRENCY CODE |
|---|----------------------|
| Cedi (Ghana) | GHC |
| CFA Franc BCEAO (Guinea-Bissau) | XOF |
| CFA Franc BEAC (Central African Republic) | XAF |
| CFP Franc (New Caledonia) | XPF |
| Chilean Peso (Chile) | CLP |
| Colombian Peso (Colombia) | COP |
| Comoro Franc (Comoros) | KMF |
| Convertible Marks (Bosnia And Herzegovina) | BAM |
| Cordoba Oro (Nicaragua) | NIO |
| Costa Rican Colon (Costa Rica) | CRC |
| Cuban Peso (Cuba) | CUP |
| Cyprus Pound (Cyprus) | CYP |
| Czech Koruna (Czech Republic) | CZK |
| Dalasi (Gambia) | GMD |
| Danish Krone (Denmark) | DKK |
| Denar (The Former Yugoslav Republic Of Macedonia) | MKD |
| Deutsche Mark (Germany) | DEM |
| Dirham (United Arab Emirates) | AED |
| Djibouti Franc (Djibouti) | DJF |
| Dobra (Sao Tome And Principe) | STD |
| Dominican Peso (Dominican Republic) | DOP |
| Dong (Vietnam) | VND |
| Drachma (Greece) | GRD |
| East Caribbean Dollar (Grenada) | XCD |
| Egyptian Pound (Egypt) | EGP |
| El Salvador Colon (El Salvador) | SVC |
| Ethiopian Birr (Ethiopia) | ETB |
| Euro (Europe) | EUR |
| Falkland Islands Pound (Falkland Islands) | FKP |
| Fiji Dollar (Fiji) | FJD |
| Forint (Hungary) | HUF |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| CURRENCY COUNTRY | CURRENCY CODE |
|--|----------------------|
| Franc Congolais (The Democratic Republic Of Congo) | CDF |
| French Franc (France) | FRF |
| Gibraltar Pound (Gibraltar) | GIP |
| Gold | XAU |
| Gourde (Haiti) | HTG |
| Guarani (Paraguay) | PYG |
| Guinea Franc (Guinea) | GNF |
| Guinea-Bissau Peso (Guinea-Bissau) | GWP |
| Guyana Dollar (Guyana) | GYD |
| Hong Kong Dollar (Hong Kong) | HKD |
| Hryvnia (Ukraine) | UAH |
| Iceland Krona (Iceland) | ISK |
| Indian Rupee (India) | INR |
| Iranian Rial (Islamic Republic Of Iran) | IRR |
| Iraqi Dinar (Iraq) | IQD |
| Irish Pound (Ireland) | IEP |
| Italian Lira (Italy) | ITL |
| Jamaican Dollar (Jamaica) | JMD |
| Jordanian Dinar (Jordan) | JOD |
| Kenyan Shilling (Kenya) | KES |
| Kina (Papua New Guinea) | PGK |
| Kip (Lao People's Democratic Republic) | LAK |
| Kroon (Estonia) | EEK |
| Kuna (Croatia) | HRK |
| Kuwaiti Dinar (Kuwait) | KWD |
| Kwacha (Malawi) | MWK |
| Kwacha (Zambia) | ZMK |
| Kwanza Reajustado (Angola) | AOR |
| Kyat (Myanmar) | MMK |
| Lari (Georgia) | GEL |
| Latvian Lats (Latvia) | LVL |

Last revised: 5/24/2011

| CURRENCY COUNTRY | CURRENCY CODE |
|---------------------------------------|----------------------|
| Lebanese Pound (Lebanon) | LBP |
| Lek (Albania) | ALL |
| Lempira (Honduras) | HNL |
| Leone (Sierra Leone) | SLL |
| Leu (Romania) | ROL |
| Lev (Bulgaria) | BGL |
| Liberian Dollar (Liberia) | LRD |
| Libyan Dinar (Libyan Arab Jamahiriya) | LYD |
| Lilangeni (Swaziland) | SZL |
| Lithuanian Litas (Lithuania) | LTL |
| Loti (Lesotho) | LSL |
| Luxembourg Franc (Luxembourg) | LUF |
| Malagasy Franc (Madagascar) | MGF |
| Malaysian Ringgit (Malaysia) | MYR |
| Maltese Lira (Malta) | MTL |
| Manat (Turkmenistan) | TMM |
| Markka (Finland) | FIM |
| Mauritius Rupee (Mauritius) | MUR |
| Metical (Mozambique) | MZM |
| Mexican Peso (Mexico) | MXN |
| Mexican Unidad de Inversion (Mexico) | MXV |
| Moldovan Leu (Republic Of Moldova) | MDL |
| Moroccan Dirham (Morocco) | MAD |
| Mvdol (Bolivia) | BOV |
| Naira (Nigeria) | NGN |
| Nakfa (Eritrea) | ERN |
| Namibia Dollar (Namibia) | NAD |
| Nepalese Rupee (Nepal) | NPR |
| Netherlands (Netherlands) | ANG |
| Netherlands Guilder (Netherlands) | NLG |
| New Dinar (Yugoslavia) | YUM |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| CURRENCY COUNTRY | CURRENCY CODE |
|--|----------------------|
| New Israeli Sheqel (Israel) | ILS |
| New Kwanza (Angola) | AON |
| New Taiwan Dollar (Province Of China Taiwan) | TWD |
| New Zaire (Zaire) | ZRN |
| New Zealand Dollar (New Zealand) | NZD |
| Ngultrum (Bhutan) | BTN |
| North Korean Won (Democratic People's Republic Of Korea) | KPW |
| Norwegian Krone (Norway) | NOK |
| Nuevo Sol (Peru) | PEN |
| Ouguiya (Mauritania) | MRO |
| Pa'anga (Tonga) | TOP |
| Pakistan Rupee (Pakistan) | PKR |
| Palladium | XPD |
| Pataca (Macau) | MOP |
| Peso Uruguayo (Uruguay) | UYU |
| Philippine Peso (Philippines) | PHP |
| Platinum | XPT |
| Portuguese Escudo (Portugal) | PTE |
| Pound Sterling (United Kingdom) | GBP |
| Pula (Botswana) | BWP |
| Qatari Rial (Qatar) | QAR |
| Quetzal (Guatemala) | GTQ |
| Rand (Financial) (Lesotho) | ZAL |
| Rand (South Africa) | ZAR |
| Rial Omani (Oman) | OMR |
| Riel (Cambodia) | KHR |
| Rufiyaa (Maldives) | MVR |
| Rupiah (Indonesia) | IDR |
| Russian Ruble (Russian Federation) | RUB |
| Russian Ruble (Russian Federation) | RUR |
| Rwanda Franc (Rwanda) | RWF |

Last revised: 5/24/2011

| CURRENCY COUNTRY | CURRENCY CODE |
|--|----------------------|
| Saudi Riyal (Saudi Arabia) | SAR |
| Schilling (Austria) | ATS |
| Seychelles Rupee (Seychelles) | SCR |
| Silver | XAG |
| Singapore Dollar (Singapore) | SGD |
| Slovak Koruna (Slovakia) | SKK |
| Solomon Islands Dollar (Solomon Islands) | SBD |
| Som (Kyrgyzstan) | KGS |
| Somali Shilling (Somalia) | SOS |
| Spanish Peseta (Spain) | ESP |
| Sri Lanka Rupee (Sri Lanka) | LKR |
| St Helena Pound (St Helena) | SHP |
| Sucre (Ecuador) | ECS |
| Sudanese Dinar (Sudan) | SDD |
| Surinam Guilder (Suriname) | SRG |
| Swedish Krona (Sweden) | SEK |
| Swiss Franc (Switzerland) | CHF |
| Syrian Pound (Syrian Arab Republic) | SYR |
| Tajik Ruble (Tajikistan) | TJR |
| Taka (Bangladesh) | BDT |
| Tala (Samoa) | WST |
| Tanzanian Shilling (United Republic Of Tanzania) | TZS |
| Tenge (Kazakhstan) | KZT |
| Timor Escudo (East Timor) | TPE |
| Tolar (Slovenia) | SIT |
| Trinidad and Tobago Dollar (Trinidad And Tobago) | TTD |
| Tugrik (Mongolia) | MNT |
| Tunisian Dinar (Tunisia) | TND |
| Turkish Lira (Turkey) | TRL |
| Uganda Shilling (Uganda) | UGX |
| Unidad de Valor Constante (Ecuador) | ECV |

Last revised: 5/24/2011

© 2009 Authorize.Net, a CyberSource solution

| CURRENCY COUNTRY | CURRENCY CODE |
|--------------------------------------|----------------------|
| Unidades de fomento (Chile) | CLF |
| US Dollar (Next day) (United States) | USN |
| US Dollar (Same day) (United States) | USS |
| US Dollar (United States) | USD |
| Uzbekistan Sum (Uzbekistan) | UZS |
| Vatu (Vanuatu) | VUV |
| Won (Republic Of Korea) | KRW |
| Yemeni Rial (Yemen) | YER |
| Yen (Japan) | JPY |
| Yuan Renminbi (China) | CNY |
| Zimbabwe Dollar (Zimbabwe) | ZWD |
| Zloty (Poland) | PLN |