

Advanced Integration Method (AIM) Developer Guide

Card Not Present Transactions

February 2014

Authorize.Net
a CyberSource solution

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 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®

Authorize.Net[®]
a CyberSource solution

Contents

[Recent Revisions to This Document](#) 5

Chapter 1	Introduction	6
	AIM Minimum Requirements	6
	Payment Card Industry (PCI) Data Security Standard	7
	Managing Integration Settings	7
	Features of AIM	8
	eCheck.Net	9
	PayPal Express Checkout	9
	Payment Processors	10
	North American Payment Processors	10
	Accepted Card Types	10
	Accepted Currencies	10
	European Payment Processors	10
	Asia-Pacific Processor	11
	Developer Support	12
	Software Development Kits	12

Chapter 2	Submitting Transactions	13
	Minimum Requirements	14
	Credit Card Transaction Types	16
	Authorization and Capture	17
	Authorization Only	17
	Prior Authorization and Capture	17
	Capture Only	18
	Credit (Refund)	18
	Unlinked Credit	19
	Void	19
	Visa Verification Transactions	20
	Partial Authorization Transactions	20
	Using the Merchant Interface	22

Chapter 3	Transaction Data Requirements	23
	Transaction Post Location	23
	AIM Transaction Submission API	23
	Merchant Information	24
	Transaction Information	25
	Order Information	28
	Itemized Order Information	29
	Customer Information	30
	Shipping Information	32
	Additional Shipping Information (Level 2 Data)	33
	x_tax	33
	x_freight	34
	x_duty	34
	x_tax_exempt	34
	x_po_num	35
	Cardholder Authentication	35
	Merchant-defined fields	37

Chapter 4	Transaction Response	39
	Fields in the Payment Gateway Response	41
	Response for Duplicate Transactions	46
	AIM Transaction Response Types	47
	Version 3.0	47
	Version 3.1	47
	Setting the Transaction Version	48
	Response Code Details	48
	Email Receipt	62

Chapter 5	Test Transactions	64
	Testing to Generate Specific Transaction Results	65

Appendix A	Fields by Transaction Type	67
	Minimum Required Fields	67
	Required Fields for Additional AIM Features	68
	Best Practice Fields	69

Appendix B	Alphabetized List of API Fields	70
-------------------	--	-----------

Recent Revisions to This Document

Date	Revision
February 2014	<p>Added x_market_type API field. See "Transaction Information," page 25.</p> <p>Added a note recommending POST instead of GET. See "Transaction Post Location," page 23.</p> <p>Updated Developer Support section. See "Developer Support," page 12.</p> <p>Added a section on PayPal Express Checkout through AIM. See "PayPal Express Checkout," page 9.</p>
October 2013	<p>Added AUD and NZD to the , "x_currency_code" field.</p> <p>Updated the table of "Payment Processors."</p>
May 2013	<p>Added EUR to the , "x_currency_code" field.</p> <p>Updated the table of "Payment Processors."</p>
April 2013	<p>Updated the table of "Payment Processors."</p>
February 2013	<p>Updated the table of "Payment Processors."</p>
October 2012	<p>Added list of "Payment Processors." and currencies, along with associated required fields.</p> <p>Added the , "x_currency_code" field.</p>

Introduction

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

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

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

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

AIM allows merchants the highest degree of customization and control over their customer checkout experience.

**Note**

For merchants who prefer a payment solution that collects, transmits, and stores cardholder data, Authorize.Net recommends the Server Integration Method (SIM). See the *SIM Developer Guide* at: http://www.authorize.net/support/SIM_guide.pdf.

SIM does not require merchants to purchase and install an SSL digital certificate, which reduces the complexity of securely handling and storing cardholder information, simplifying compliance with the Payment Card Industry (PCI) Data Security Standard.

AIM Minimum Requirements

Before you begin, check with the merchant to make sure that the following AIM requirements have been met. It is strongly recommended that you work closely with the merchant to ensure that any other business and Web site requirements (for example,

bank or processor requirements, Web site design preferences) are included in their AIM integration.

- The merchant must have a merchant bank account that allows Internet transactions.
- The merchant must have an e-commerce (Card Not Present) Authorize.Net Payment Gateway account.
- The merchant must have a valid Secure Sockets Layer (SSL) certificate and their Web site must be capable of initiating both client- and server-side SSL connections.
- The merchant's website must have server-side scripting or CGI capabilities such as ASP Classic, C#, Cold Fusion, Java, Perl, PHP or VB.Net.
- The merchant must be able to store payment gateway account data securely (for example, API Login ID, or Transaction Key).

Payment Card Industry (PCI) Data Security Standard



Important

AIM involves transmitting sensitive cardholder data using the merchant's Web server. Therefore, if the merchant stores cardholder information, it must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. For more information about PCI and other industry standard processing practices, see the [Developer Security Best Practices White Paper](#).

If merchants need a solution that collects, transmits, and stores cardholder data, they should use the Server Implementation Method (SIM). For more information about SIM, see the [SIM Developer Guide](#).

Managing Integration Settings

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

- Included in the transaction request on a per-transaction basis using the application programming interface (API), as described in this guide.

- Configured in the Merchant Interface and applied to all transactions.



Important

The Merchant Interface at <https://account.authorize.net> is a secure website on which merchants can manage their payment gateway account settings, including their Web site integration settings. Review the *Merchant Integration Guide* at <http://www.authorize.net/support/merchant/> for information on managing the merchant's payment gateway integration using the Merchant Interface.

Transaction settings submitted in the transaction request will override transaction settings configured in the Merchant Interface. However, be aware that some integration settings *must* be configured in the Merchant Interface. To help the merchant maintain a robust integration, review with the merchant the integration settings that can be configured in the Merchant Interface and determine which settings can be posted on a per-transaction basis and which should be configured in the Merchant Interface. See [Appendix A, "Fields by Transaction Type," on page 67](#) for a list of fields that can be submitted on a per-transaction basis.

Features of AIM

In addition to basic transaction processing, AIM provides merchants with several features for configuring transaction security options and further customizing their customers' checkout experience. These features are listed in the AIM feature selection table below. Take a few moments to discuss these features with your merchant and select the appropriate features for their integration.

Table 1 Features of AIM

Features	Description	Requirements
Address Verification Service (AVS) Filter	This feature enables merchants to compare the billing address submitted by the customer for the transaction with the address on file at the card issuing bank. Filter settings in the Merchant Interface allow the merchant to reject transactions based on the AVS response received.	To implement AVS, the merchant must require the Address and ZIP Code fields on their custom payment form. For more information about AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ .

Table 1 Features of AIM (Continued)

Features	Description	Requirements
Card Code Verification (CCV) Filter	This feature enables merchants to compare the card code submitted by the customer for the transaction with the card code on file at the card issuing bank. Filter settings in the Merchant Interface allow the merchant to reject transactions based on the CCV response received.	To implement CCV, the merchant must require the Card Code on their custom payment form. For more information about CCV, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ .
Itemized Order Information	This feature enables merchants to submit details for items purchased. This information is included in the customer's confirmation email, in the Transaction Details for the transaction, and in QuickBooks download reports in the Merchant Interface.	To implement Itemized Order Information, the line item field must be submitted for each transaction. See " Itemized Order Information ," page 29 for details.
Email Receipt	This feature enables merchants to have the payment gateway send an automatic email receipt to their customers.	To configure the payment gateway email receipt, merchants must require the customer email addresses on their custom payment form, and settings must be configured in the Email Receipts section of the Settings menu in the Merchant Interface or submitted for each transaction. See " Email Receipt ," page 62 for details.

eCheck.Net

In addition to processing credit card transactions, the payment gateway also supports electronic check transactions with our exclusive eCheck.Net[®] product. Contact the merchant to determine whether eCheck.Net is enabled for their payment gateway account or whether they would like to have it enabled. If eCheck.Net is enabled, you must ensure that the merchant's Web site integration supports all eCheck.Net field requirements. See the [eCheck.Net Developer Guide](#) for more information.

PayPal Express Checkout

You can use AIM to implement PayPal Express Checkout as an alternative payment. PayPal Express Checkout for AIM is currently in beta. Contact your Authorize.Net sales representative for more information.

Payment Processors

The currencies that a merchant can accept through Authorize.Net are determined by their payment processor.

North American Payment Processors

Accepted Card Types

All North American payment processors supported by Authorize.Net accept the following card types:

- American Express
- Diners Club
- Discover
- JCB
- Mastercard
- Visa

Accepted Currencies

Table 2 North American Payment Processors and Accepted Currencies

Payment Processor	Accepted Currencies
Chase Paymentech Tampa Processing Platform	United States Dollar (USD) Canadian Dollar (CAD)
Elavon	United States Dollar (USD) Canadian Dollar (CAD)
First Data Merchant Services (FDMS) Omaha, Nashville, and EFSNet Processing Platforms	United States Dollar (USD)
Global Payments	United States Dollar (USD) Canadian Dollar (CAD)
Heartland Payment Systems	United States Dollar (USD)
TSYS Acquiring Solutions	United States Dollar (USD)
WorldPay Atlanta Processing Platform	United States Dollar (USD)

European Payment Processors

The following European payment processors are supported by Authorize.Net for Card Not

Present (CNP) transactions.

Table 3 European Payment Processors, Accepted Card Types, and Accepted Currencies

Payment Processor	Accepted Card Types	Accepted Currencies
AIB Merchant Services	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	British Pounds (GBP) Euro (EUR) United States Dollar (USD)
Barclaycard	<ul style="list-style-type: none"> ■ JCB ■ Mastercard ■ Visa 	British Pounds (GBP) Euro (EUR)
First Data Merchant Solutions (MSIP platform)	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	British Pounds (GBP)
HSBC Merchant Services	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	British Pounds (GBP) Euro (EUR)
Lloyds Bank Cardnet	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	British Pounds (GBP)
Streamline	<ul style="list-style-type: none"> ■ JCB ■ Mastercard ■ Visa 	British Pounds (GBP) Euro (EUR) United States Dollar (USD)

Asia-Pacific Processor

The following Asia-Pacific payment processor is supported by Authorize.Net for Card Not Present (CNP) transactions.

Table 4 Asia-Pacific Payment Processor, Accepted Card Types, and Accepted Currencies

Payment Processor	Accepted Card Types	Accepted Currencies
FDI Australia	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	Australian Dollar (AUD) New Zealand Dollar (NZD) United States Dollar (USD)
Westpac	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	Australian Dollar (AUD)

For information on setting the currency using the AIM API, see "[x_currency_code](#)".

Developer Support

Several resources are available to help you successfully integrate a merchant Web site or other application to the Authorize.Net payment gateway.

- The Developer Center at <http://developer.authorize.net> provides sandbox accounts, sample code, FAQs, and troubleshooting tools.
- Developer Training Videos are available at <http://developer.authorize.net/training>.
- The Developer Community at <http://community.developer.authorize.net> provides answers to questions from other Authorize.Net developers.
- Developer Support is available at <http://developer.authorize.net/support>.

If you have suggestions for improving or correcting this guide, send email to documentation@authorize.net.

Software Development Kits

Authorize.Net offers software development kits (SDKs) that present an alternate object-oriented model, in several popular languages. To use these SDKs, the merchant's transaction version must be set to 3.1. The SDK performs the core payment activities (such as error handling and parsing, network communication, and data encoding) behind the scenes.

The SDK provides utilities to help developers build payment flows for each of the integration methods. You can download the SDKs at <http://developer.authorize.net/downloads/>.

Submitting Transactions

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

To implement AIM for a merchant's Web site or proprietary business application, you will need to develop an application that:

- Securely obtains all of the information required to process a transaction (including data requirements specified by the merchant).
- Initiates an SSL connection from the merchant's Web server to the payment gateway transaction post location to pass transaction data in name/value pairs.
- Receives and parses the transaction response from the payment gateway and displays the results to the customer.

You can develop the application in one of two ways:

- by yourself using the information provided in this document
- use Authorize.Net sample code available for free from our [Developer Center](#).

If you choose to use sample code, be aware that to achieve a successful implementation, the code must be modified with the merchant's specific payment gateway account information. Be sure to carefully review the readme.txt files and comments included in each file of sample code in order to achieve a faster, successful integration.

Developer test accounts with API Login IDs and Transaction Keys are also available to test your integration methods to the Authorize.Net Payment Gateway at <http://developer.authorize.net/testaccount>.

Minimum Requirements

The following table represents the minimum fields required for submitting a credit card transaction request to the payment gateway using AIM. The data fields are name/value pairs with the following syntax:

```
x_name_of_field=value of field
```

Table 5 Minimum AIM Fields

Field Name	Description
x_login	<p>Value: The merchant's unique API Login ID</p> <p>Format: Up to 20 characters</p> <p>Notes: The merchant API Login ID is provided in the Merchant Interface and must be stored securely.</p> <p>The API Login ID and Transaction Key together provide the merchant the authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ for more information.</p>
x_tran_key	<p>Value: The merchant's unique Transaction Key</p> <p>Format: 16 characters</p> <p>Notes: The merchant Transaction Key is provided in the Merchant Interface and must be stored securely.</p> <p>The API Login ID and Transaction Key together provide the merchant the authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ for more information.</p>
x_type	<p>Value: The type of credit card transaction</p> <p>Format: AUTH_CAPTURE (default), AUTH_ONLY, CAPTURE_ONLY, CREDIT, PRIOR_AUTH_CAPTURE, VOID</p> <p>Notes: If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an AUTH_CAPTURE.</p>
x_amount	<p>Value: The amount of the transaction</p> <p>Format: Up to 15 digits with a decimal point (no dollar symbol). For example, 8.95.</p> <p>Notes: This is the total amount and must <i>include</i> tax, shipping, and any other charges.</p>

Table 5 Minimum AIM Fields (Continued)

Field Name	Description
x_card_num	<p>Value: The customer's credit card number</p> <p>Format; Between 13 and 16 digits without spaces. When x_type=CREDIT, only the last four digits are required.</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the Developer Security Best Practices White Paper.</p>
x_exp_date	<p>Value: The customer's credit card expiration date.</p> <p>Format; MMY, MMYY, MM/YY,MM-YY, MMYYYY, MM/YYYY, MM-YYYY</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the Developer Security Best Practices White Paper.</p>
x_trans_id	<p>Value: The payment gateway-assigned transaction ID of an original transaction</p> <p>Notes: Required only for CREDIT, PRIOR_AUTH_CAPTURE, and VOID transactions. Do not include this field if you are providing the x_split_tender_id field.</p> <p>For more information about transaction types, see "Credit Card Transaction Types," page 16.</p>
x_split_tender_id	<p>Value: The payment gateway-assigned ID that links the current authorization request to the original authorization request.</p> <p>Format; Numeric</p> <p>Notes: This value applies only to partial authorization transactions, and is returned in the reply message from the original authorization request.</p> <p>For more information, see "Partial Authorization Transactions," page 20.</p>
x_auth_code	<p>Value: The authorization code of an original transaction not authorized on the payment gateway</p> <p>Format; 6 characters</p> <p>Notes: Required only for CAPTURE_ONLY transactions. See "Credit Card Transaction Types," page 16".</p>
x_relay_response	<p>Value: FALSE</p> <p>Format; FALSE, F, NO, N</p> <p>Notes: SIM applications use relay response. Set this to false if you are using AIM.</p>

Table 5 Minimum AIM Fields (Continued)

Field Name	Description
x_delim_data	<p>Value: Indicates whether a delimited transaction response is required</p> <p>Format: TRUE, T, YES, Y</p> <p>Notes: A value of TRUE indicates a request for delimited response from the payment gateway. Since all AIM transactions are direct response, a value of TRUE is required.</p> <p>Submit this field for each transaction to be sure that transaction responses are returned in the correct format.</p> <p>See Chapter 4, "Transaction Response," on page 39 for more about delimited response.</p>

**Note**

When using a European Payment Processor, additional fields are required. For more information, see ["Customer Information"](#).

Credit Card Transaction Types

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

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

The payment gateway supports the following credit card transaction types.

**Note**

Some of the field requirements listed in this section for each credit card transaction type are in addition to the minimum field requirements already set forth above for ALL transactions submitted to the payment gateway. For a list of all fields that are required for each credit card transaction type, see [Appendix A, "Fields by Transaction Type," on page 67](#).

Authorization and Capture

This is the most common type of credit card transaction and is the default payment gateway transaction type. The amount is sent for authorization, and if approved, is automatically submitted for settlement.

The unique field requirement for an Authorization and Capture transaction is:

```
x_type=AUTH_CAPTURE
```

Authorization Only

This transaction type is sent for authorization only. The transaction will not be sent for settlement until the credit card transaction type Prior Authorization and Capture (see definition below) is submitted, or the transaction is submitted for capture manually in the Merchant Interface. For more information about capturing Authorization Only transactions in the Merchant Interface, see the *Merchant Integration Guide* at <http://www.authorize.net/support/merchant/>.

If action for the Authorization Only transaction is not taken on the payment gateway within 30 days, the authorization expires and is no longer available for capture. A new Authorization Only transaction would then have to be submitted to obtain a new authorization code.

The unique field requirement for an Authorization Only transaction is:

```
x_type=AUTH_ONLY
```

Merchants can submit Authorization Only transactions if they want to verify the availability of funds on the customer's credit card before finalizing the transaction. This transaction type can also be submitted if the merchant does not currently have an item in stock or wants to review orders before shipping goods.

Prior Authorization and Capture

This transaction type is used to complete an Authorization Only transaction that was successfully authorized through the payment gateway.

**Note**

An Authorization Only and a Prior Authorization and Capture together are considered one complete transaction. After the Prior Authorization and Capture is submitted, the transaction will be sent for settlement.

The payment gateway accepts this transaction type and initiates settlement if the following conditions are met:

- The original Authorization Only transaction was submitted within the previous 30 days (Authorization Only transactions expire on the payment gateway after 30 days).
- The transaction is submitted with the valid transaction ID (*x_trans_id*) of an original, successfully authorized, Authorization Only transaction.
- The original transaction is not yet captured, expired or generated an error.
- The amount being requested for capture is less than or equal to the original authorized amount. Only a single Prior Authorization and Capture transaction can be submitted against an Authorization Only.

The unique field requirements for a Prior Authorization and Capture are:

```
x_type=PRIOR_AUTH_CAPTURE
x_trans_id=Transaction ID here
```

For this transaction type, the amount field (*x_amount*) is required only if a Prior Authorization and Capture is submitted for an amount that is *less* than the amount of the original Authorization Only transaction. If no amount is submitted, the payment gateway will initiate settlement for the amount of the original authorized transaction.

Capture Only

This transaction type is used to complete a previously authorized transaction that was *not* originally submitted through the payment gateway or that requires voice authorization.

The payment gateway accepts this transaction type and initiates settlement if the following conditions are met:

- The transaction is submitted with the valid authorization code issued to the merchant to complete the transaction.

The unique field requirements for a Capture Only transaction are:

```
x_type=CAPTURE_ONLY
x_auth_code=Authorization Code here
```

For instructions on how to perform a Capture Only transaction in the Merchant Interface, see the *Merchant Integration Guide* at <http://www.authorize.net/support/merchant/>.

Credit (Refund)

This transaction type is used to refund a customer for a transaction that was originally processed and successfully settled through the payment gateway.

The payment gateway accepts Credits if the following conditions are met:

- The transaction is submitted with the valid transaction ID (*x_trans_id*) of an original, successfully *settled* transaction.
- The amount being requested for refund is less than or equal to the original settled amount.
- The sum of multiple Credit transactions submitted against the original transaction is less than or equal to the original settled amount.
- At least the last four digits of the credit card number (*x_card_num*) used for the original, successfully settled transaction are submitted. An expiration date is not required.
- The transaction is submitted within 120 days of the settlement date of the original transaction.

The unique field requirements for a Credit transaction are:

```
x_type=CREDIT
```

```
x_trans_id=Transaction ID here
```

```
x_card_num=Full credit card number or last four digits only here
```

Unlinked Credit

This transaction type is used to issue a refund for a transaction that was *not* originally submitted through the payment gateway. It also enables the merchant to override restrictions for submitting refunds for payment gateway transactions, for example, if the merchant is beyond the 120-day period for submitting a refund or would like to refund an amount that is greater than the original transaction amount.

The ability to submit unlinked credits is not a standard feature of a merchant's payment gateway account. To be enabled for expanded credits capability (ECC), the merchant must submit an application, which can be found at <http://www.authorize.net/files/ecc.pdf>.



A transaction ID must not be submitted with an Unlinked Credit. If ECC is enabled for the merchant's account, and a transaction ID is submitted with the Unlinked Credit transaction, then the payment gateway will attempt to apply the credit to an original transaction with the transaction ID submitted.

The unique field requirement for an Unlinked Credit is:

```
x_type=CREDIT
```

Void

This transaction type can be used to cancel either an original transaction that is not yet settled, or an entire order composed of more than one transactions. A Void prevents the

transaction or the order from being sent for settlement. A Void can be submitted against any other transaction type.

**Note**

If you are not sure whether a transaction is settled, you can attempt to submit a Void first. If the Void transaction results in an error, the original transaction is likely settled and you can submit a Credit for the transaction.

The payment gateway accepts Voids if the following conditions are met:

- The transaction is submitted with the valid transaction ID (*x_trans_id*) of an original, successfully authorized transaction. To void an entire order, submit the split tender ID (*x_split_tender_id*).
- The original transaction is not already settled, expired or generated an error.

The unique field requirements for a Void transaction are:

```
x_type=VOID
```

```
x_trans_id=Transaction ID, or x_split_tender_id=Split Tender ID.
```

**Note**

Typically, Authorization Only or Authorization and Capture are the primary transaction types submitted by an e-commerce Web site or other application. Although they most likely will not be used for the merchant's website integration, all other transaction types listed above can be integrated for automatic submission into an internal or enterprise application, like those used in a call center, or they can also be submitted by the merchant manually using the Virtual Terminal in the Merchant Interface.

Visa Verification Transactions

The following are required for \$0 Visa verification calls:

- The transaction type must be AUTH_ONLY. All other transaction types will be rejected.
- Bill To address (*x_address*) and zip code (*x_zip*) are required in order to perform the AVS check.

Not all processors accept zero dollar amount.

Partial Authorization Transactions

A split tender order is an order in which two or more transactions are used to cover the total amount of the order.

Merchants 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.

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, send a void request, using the split tender ID instead of a transaction ID. This action will void all the transactions in the group.

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, in addition to the Transaction ID. The split tender ID is used on subsequent payments for that purchase.
- The transaction is not submitted for settlement until either the merchant submits payments adding up to the full requested amount, or until the merchant indicates that the transaction has been completed (when all or part of the remaining balance is paid in cash).
- You can void all transactions in an order using a split tender ID, or you can void individual transactions using a transaction 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. This parameter is sent to the merchant during relay response processing. It is not included in the delimited response.
- `x_card_type` (output)—the card type.

Using the Merchant Interface

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

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

Transaction Data Requirements

The standard payment gateway Application Programming Interface (API) consists of required information fields (introduced in the previous section) and additional optional fields that can be submitted to the payment gateway for real-time transaction processing.

Transaction Post Location

The merchant's Web site should POST transaction requests to the following payment gateway URL:

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

**Note**

If you are using an Authorize.Net developer test account, test transactions are posted to a staging environment at <https://test.authorize.net/gateway/transact.dll>. If you do not have a developer test account, you can sign up for one at <http://developer.authorize.net>.

**Note**

Transactions should be sent using HTTP POST, not HTTP GET. HTTP GET sends information in clear text and is therefore not secure.

For more information, see [RFC 2616, section 15.1.3](#).

AIM Transaction Submission API

The following tables list the transaction data fields that can be submitted using the transaction request string. Some of these fields can also be configured in the Merchant Interface. For more information about configuring these settings in the Merchant Interface, 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 Information

Table 6 Merchant Information

Field Name	Description
x_login	<p>Required</p> <p>Value: The merchant's unique API Login ID</p> <p>Format: Up to 20 characters</p> <p>Notes: The merchant API Login ID is provided in the Merchant Interface and must be stored securely.</p> <p>The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ for more information.</p>
x_tran_key	<p>Required</p> <p>Value: The merchant's unique Transaction Key</p> <p>Format: 16 characters</p> <p>Notes: The merchant Transaction Key is provided in the Merchant Interface and must be stored securely.</p> <p>The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ for more information.</p>
x_allow_partial_Auth	<p>Optional</p> <p>Value: True, False</p> <p>Notes: Indicates if the transaction is enabled for partial authorization. Including this field in the transaction request overrides your account configuration.</p> <p>For more information, see "Partial Authorization Transactions," page 20.</p>

Transaction Information

Table 7 Transaction Information

Field Name	Description
x_version	<p>Required</p> <p>Value: The merchant's transaction version</p> <p>Format: 3.0, 3.1</p> <p>Notes: Indicates to the system the set of fields that will be included in the response:</p> <p>3.0 is the default version.</p> <p>3.1 allows the merchant to utilize the Card Code feature and the Partial Authorization feature, and is the current standard version.</p> <p>It is highly recommended that you submit this field on a per-transaction basis to be sure that the formats of transaction requests and the responses you receive are consistent.</p> <p>For more information, see Appendix A, "Fields by Transaction Type," on page 67 of this document.</p>
x_type	<p>Optional</p> <p>Value: The type of credit card transaction</p> <p>Format: AUTH_CAPTURE (default), AUTH_ONLY, CAPTURE_ONLY, CREDIT, PRIOR_AUTH_CAPTURE, VOID</p> <p>Notes: If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an AUTH_CAPTURE.</p>
x_method	<p>Optional</p> <p>Value: The payment method</p> <p>Format: CC or ECHECK</p> <p>Notes: The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If this field is not submitted or is blank, the value will default to CC.</p> <p>For more information about eCheck.Net transaction requirements, see the eCheck.Net Developer Guide.</p>
x_market_type	<p>Optional</p> <p>Value: One of the following.</p> <p>0 for e-commerce</p> <p>1 for moto</p> <p>2 for retail.</p> <p>Notes: If your account type is Card Present, the default is 2 and only 2 can be used. If your account type is blended, the default is 0, but x_market_type can be overridden.</p>

Table 7 Transaction Information (Continued)

Field Name	Description
x_recurring_billing	<p>Optional</p> <p>Value: The recurring billing status</p> <p>Format: TRUE, FALSE,T, F, YES, NO,Y, N,1, 0</p> <p>Notes: Indicating marker used by merchant account providers to identify transactions which originate from merchant hosted recurring billing applications. This value is not affiliated with Automated Recurring Billing.</p>
x_amount	<p>Required</p> <p>Value: The amount of the transaction</p> <p>Format: Up to 15 digits with a decimal point (no dollar symbol). For example, 8.95</p> <p>Notes: This is the total amount and must include tax, shipping, and any other charges. The amount can either be hard coded or posted to a script.</p>
x_currency_code	<p>Optional</p> <p>Value: AUD, USD, CAD, EUR, GBP, or NZD.</p> <p>Format: 3 character string.</p> <p>Notes: If this field is not submitted, the default is the currency selected by the merchant's payment processor. Setting this field to a currency that is not supported by the payment processor will result in an error.</p>
x_card_num	<p>Required</p> <p>Value: The customer's credit card number</p> <p>Format: Between 13 and 16 digits without spaces. When x_type=CREDIT, only the last four digits are required.</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the Developer Security Best Practices White Paper.</p>
x_exp_date	<p>Required</p> <p>Value: The customer's credit card expiration date</p> <p>Format: MMY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the Developer Security Best Practices White Paper.</p>

Table 7 Transaction Information (Continued)

Field Name	Description
x_card_code	<p>Optional</p> <p>Value: The three- or four-digit number on the back of a credit card (on the front for American Express).</p> <p>Format: Numeric</p> <p>Notes: This field is required if the merchant would like to use the Card Code Verification (CCV) security feature. For more information, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant.</p> <p>Cardholder information must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>See the <i>Developer Security Best Practices White Paper</i> for more information.</p>
x_trans_id	<p>Conditional</p> <p>Required only for CREDIT, PRIOR_AUTH_CAPTURE, and VOID transactions</p> <p>Value: The payment gateway assigned transaction ID of an original transaction</p> <p>For more information about transaction types, see the "Credit Card Transaction Types," page 16 section of this document.</p>
x_split_tender_id	<p>Conditional</p> <p>Value: The payment gateway-assisted 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.</p> <p>Notes: 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 will be 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.</p> <p>For more information about partial authorization transactions, see the "Partial Authorization Transactions," page 20 topic.</p>
x_auth_code	<p>Conditional</p> <p>Required only for CAPTURE_ONLY transactions</p> <p>Value: The authorization code of an original transaction <i>not</i> authorized on the payment gateway</p> <p>Format: 6 characters</p> <p>Notes: See the "Credit Card Transaction Types," page 16" section of this document.</p>

Table 7 Transaction Information (Continued)

Field Name	Description
x_test_request	<p>Optional</p> <p>Value: The request to process test transactions</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates if the transaction should be processed as a test transaction.</p> <p>See Chapter 5, "Test Transactions," on page 64 of this guide for more information.</p>
x_duplicate_window	<p>Optional</p> <p>Value: The window of time after the submission of a transaction that a duplicate transaction can not be submitted</p> <p>Format: Any value between 0 and 28800 (no comma)</p> <p>Notes: 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).</p> <p>If a value less than 0 is sent, the payment gateway will default to 0 seconds. If a value greater than 28800 is sent, the payment gateway will default to 28800. If no value is sent, the payment gateway will default to 2 minutes (120 seconds).</p> <p>If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See the "Response for Duplicate Transactions," page 46 section of this guide for more information.</p>

Order Information

Table 8 Order Information

Field Name	Description
x_invoice_num	<p>Optional</p> <p>Value: The merchant-assigned invoice number for the transaction</p> <p>Format: Up to 20 characters (no symbols)</p> <p>Notes: The invoice number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>

Table 8 Order Information (Continued)

Field Name	Description
x_description	<p>Optional</p> <p>Value: The transaction description</p> <p>Format: Up to 255 characters (no symbols)</p> <p>Notes: The description must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>

Itemized Order Information

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

The value for the x_line_item field is capable of including delimited item information. Item Information must be delimited by a bracketed pipe<|>. Line item values must be included in the order listed below.

The following table describes the Item Information elements of the x_line_item field. A code example is presented after the table.

Table 9 Delimited x_line_item Information

Item Information Elements	Description
Item ID< >	<p>Required</p> <p>Value: The ID assigned to an item.</p> <p>Format: Up to 31 characters</p>
Item Name< >	<p>Required</p> <p>Value: The name of an item.</p> <p>Format: Up to 31 characters</p>
Item Description< >	<p>Optional</p> <p>Value: A detailed description of an item.</p> <p>Format: Up to 255 characters</p>
Item Quantity< >	<p>Required</p> <p>Value: The quantity of the item on this order.</p> <p>Format: Up to two decimal places. Must be a positive number</p>

Table 9 Delimited x_line_item Information (Continued)

Item Information Elements	Description
Item Price (unit cost)< >	Required Value: Cost of an item per unit, <i>excluding</i> tax, freight, and duty. Format: Up to two decimal places. Must be a positive number. The dollar sign (\$) is not allowed when submitting delimited information.
Item Taxable	Optional (FALSE by default) Value: Indicates whether the item is subject to tax. Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0

The merchant can submit up to 30 distinct line items containing itemized order information per transaction. All field separators are required whether there the field has a value a value. In the example below, the Item Description field after *golf balls*</> has no value, yet the bracketed pipe remains.

Example Submitting itemized order information

```
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 Authorization and 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 applies.

Customer Information

Table 10 Customer Information

Field Name	Description
x_first_name	Required only when using a European Payment Processor. Value: The first name associated with the customer's billing address Format: Up to 50 characters (no symbols)
x_last_name	Required only when using a European Payment Processor. Value: The last name associated with the customer's billing address Format: Up to 50 characters (no symbols)

Table 10 Customer Information (Continued)

Field Name	Description
x_company	Optional Value: The company associated with the customer's billing address Format: Up to 50 characters (no symbols)
x_address	Required only when using a European Payment Processor. Value: The customer's billing address Format: Up to 60 characters (no symbols) Required if the merchant would like to use the Address Verification Service security feature. For more information on AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ .
x_city	Required only when using a European Payment Processor. Value: The city of the customer's billing address Format: Up to 40 characters (no symbols)
x_state	Required only when using a European Payment Processor. Value: The state of the customer's billing address Format: Up to 40 characters (no symbols) or a valid two-character state code
x_zip	Required only when using a European Payment Processor. Value: The ZIP code of the customer's billing address Format: Up to 20 characters (no symbols) Required if the merchant would like to use the Address Verification Service security feature. For more information on AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ .
x_country	Required only when using a European Payment Processor. Value: The country of the customer's billing address Format: Up to 60 characters (no symbols)
x_phone	Optional Value: The phone number associated with the customer's billing address Format: Up to 25 digits (no letters). For example, (123)123-1234
x_fax	Optional Value: The fax number associated with the customer's billing address Format: Up to 25 digits (no letters). For example, (123)123-1234

Table 10 Customer Information (Continued)

Field Name	Description
x_email	<p>Required only when using a European Payment Processor.</p> <p>Value: The customer's valid email address</p> <p>Format: Up to 255 characters. For example, janedoe@customer.com</p> <p>Notes: The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p> <p>For more information about Email Receipts, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_cust_id	<p>Optional</p> <p>Value: The merchant assigned customer ID</p> <p>Format: Up to 20 characters (no symbols)</p> <p>Notes: The unique identifier to represent the customer associated with the transaction.</p> <p>The customer ID must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>
x_customer_ip	<p>Optional</p> <p>Value: The customer's IP address</p> <p>Format: Up to 15 characters (no letters). For example, 255.255.255.255</p> <p>Notes: IP address of the customer initiating the transaction. If this value is not passed, it will default to 255.255.255.255.</p> <p>This field is required when using customer-IP based Advanced Fraud Detection Suite™ (AFDS) filters. For more information about AFDS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>

Shipping Information

Table 11 Shipping Information

Field Name	Description
x_ship_to_first_name	<p>Optional</p> <p>Value: The first name associated with the customer's shipping address</p> <p>Format: Up to 50 characters (no symbols)</p>

Table 11 Shipping Information (Continued)

Field Name	Description
x_ship_to_last_name	Optional Value: The last name associated with the customer's shipping address Format: Up to 50 characters (no symbols)
x_ship_to_company	Optional Value: The company associated with the customer's shipping address Format: Up to 50 characters (no symbols)
x_ship_to_address	Optional Value: The customer's shipping address Format: Up to 60 characters (no symbols)
x_ship_to_city	Optional Value: The city of the customer's shipping address Format: Up to 40 characters (no symbols)
x_ship_to_state	Optional Value: The state of the customer's shipping address Format: Up to 40 characters (no symbols) or a valid two-character state code
x_ship_to_zip	Optional Value: The ZIP code of the customer's shipping address Format: Up to 20 characters (no symbols)
x_ship_to_country	Optional Value: The country of the customer's shipping address Format: Up to 60 characters (no symbols)

Additional Shipping Information (Level 2 Data)

The following sections describe shipping information field names and their child elements. Delimited tax, freight, and duty information are not returned in the transaction response or in the merchant confirmation email. This information is displayed only on the Transaction Detail page in the Merchant Interface.

x_tax

This optional field can contain either the valid tax amount or the delimited tax information. When submitting delimited tax information, values must be delimited by a bracketed pipe <|> in the order shown below. The total amount of the transaction in x_amount must include this amount.

The delimited tax information elements are:

- tax item name<|>
- tax description<|>
- tax amount: The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must include this amount.

Example: x_tax=Tax1<|>state tax<|>0.09

x_freight

This optional field can contain either the valid freight amount, or delimited freight information. When submitting delimited freight information, values must be delimited by a bracketed pipe <|>, as shown in the example below. The total amount of the transaction in x_amount must include this amount.

The delimited freight information elements are:

- tax item name<|>
- tax description<|>
- tax amount: The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must include this amount.

Example: x_freight=Freight<|>ground overnight<|>12.95

x_duty

This optional field can contain either the valid duty amount or delimited duty information. When submitting delimited duty information, values must be delimited by a pipe <|>, as shown in the example below. The total amount of the transaction in x_amount must include this amount.

The delimited duty information elements are:

- duty item name<|>
- freight description<|>
- freight amount: The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must include this amount.

Example: x_duty=Duty1<|>export<|>15.00

x_tax_exempt

This optional field can contain the tax exempt status of the order.

The values of this field can include: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0

x_po_num

This optional field can contain the merchant-assigned purchase order number, up to 25 characters, no symbols. The purchase order number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.

Cardholder Authentication

The payment gateway supports the transmission of authentication fields for the following cardholder authentication programs:

- Verified by Visa
- MasterCard® SecureCode™

Merchants using a third-party cardholder authentication solution can submit the following authentication values with Visa and/or MasterCard transactions.



Note

The cardholder authentication fields are currently supported only through the Chase Paymentech, FDMS Nashville, Global Payments and TSYS processors for Visa and MasterCard transactions. If cardholder authentication information is submitted for transactions processed through any other processor, it will be ignored.

Table 12 Cardholder Authentication Fields

Field Name	Description
x_authentication_indicator	<p>Optional</p> <p>Value: The electronic commerce indicator (ECI) value for a Visa transaction; <i>or</i> the universal cardholder authentication field indicator (UCAFI) for a MasterCard transaction obtained by the merchant after the authentication process.</p> <p>Format: Special characters included in this value must be URL encoded.</p> <p>Notes: Required only for AUTH_ONLY and AUTH_CAPTURE transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored.</p> <p>This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.</p>

Table 12 Cardholder Authentication Fields (Continued)

Field Name	Description
x_cardholder_authentication_value	<p data-bbox="803 310 893 338">Optional</p> <p data-bbox="803 359 1421 510">Value: The cardholder authentication verification value (CAVV) for a Visa transaction; <i>or</i> accountholder authentication value (AVV)/ universal cardholder authentication field (UCAF) for a MasterCard transaction obtained by the merchant after the authentication process.</p> <p data-bbox="803 531 1421 590">Format: Special characters included in this value must be URL encoded.</p> <p data-bbox="803 611 1421 730">Notes: Required only for AUTH_ONLY and AUTH_CAPTURE transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored.</p> <p data-bbox="803 751 1421 835">This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.</p>

Invalid combinations of the *x_authentication_indicator* and *x_cardholder_authentication_value* fields will cause the transaction to generate an error.

Valid value combinations for these fields are as follows:

Visa

Table 13 Valid Value Combinations For Verified By Visa Fields

Authentication Indicator	Cardholder Authentication Value
5	Not null
6	Not null
6	Null/Blank
7	Null/Blank
7	Not null (some international issuers can provide a CAVV value when ECI is 7)
Null/Blank	Null/Blank

MasterCard

Table 14 Valid Value Combinations For MasterCard SecureCode Fields

Authentication Indicator	Cardholder Authentication Value
0	Blank /Null
2	Not null
1	Null
Null	Null

For example, when the MasterCard value for *x_authentication_indicator* is "1," the value for *x_cardholder_authentication_value* must be null. In this scenario, if a value is submitted for *x_cardholder_authentication_value*, the transaction fails validation and is rejected.

The authentication verification value returned by Visa or MasterCard is included in the transaction response from the payment gateway and is also included on the Transaction Detail page for the transaction in the Merchant Interface.

Merchant-defined fields

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

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

**Note**

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

**Warning**

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

Transaction Response

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.

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 that can be displayed 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.

You can use the following fields to customize the format of the payment gateway transaction response. You can also configure these settings in the Merchant Interface. For more information about configuring these settings in the Merchant Interface, 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&
```

Table 15 Response Request Fields

Field Name	Description
x_delim_data	<p>Value: The request to receive a delimited transaction response</p> <p>format: TRUE</p> <p>Notes: In order to receive a delimited response from the payment gateway, submit this field with a value of TRUE, or the merchant has to configure a delimited response through the Merchant Interface.</p> <p>Submit this field for each transaction basis to be sure that transaction responses are returned in the correct format.</p>

Table 15 Response Request Fields (Continued)

Field Name	Description
x_delim_char	<p>Value: The delimiting character</p> <p>Format: A single symbol. For example:</p> <ul style="list-style-type: none"> , (comma) (pipe) " (double quotes) ' (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star) <p>Notes: The character used to separate fields in the transaction response. The payment gateway uses the character passed in this field or, if no value is passed, the value stored in the Merchant Interface.</p> <p>If this field is passed, and the value is null, it overrides the value stored in the Merchant Interface and there is no delimiting character in the transaction response.</p> <p>Submit this field for each transaction to be sure that transaction responses are returned in the correct format.</p>
x_encap_char	<p>Value: The encapsulating character</p> <p>Format: A single symbol. For example:</p> <ul style="list-style-type: none"> (pipe) " (double quotes) ' (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star) <p>Notes: The character used to encapsulate the fields in the transaction response. This is necessary only if your delimiting character could possibly be included in any field values.</p> <p>The payment gateway uses the character passed in this field or, If no value is passed, the value stored in the Merchant Interface.</p>

Fields in the Payment Gateway Response

The following table lists the fields returned in the response from the payment gateway, in the order that they are listed in the response.

Table 16 Payment Gateway Response Fields

Order	Field Name	Description
1	Response Code	<p>Value: The overall status of the transaction</p> <p>Format:</p> <ul style="list-style-type: none"> ■ 1 = Approved ■ 2 = Declined ■ 3 = Error ■ 4 = Held for Review
2	Response Subcode	<p>Value: A code used by the payment gateway for internal transaction tracking</p>
3	Response Reason Code	<p>Value: A code that represents more details about the result of the transaction</p> <p>Format: Numeric</p> <p>Notes: See the "Response Code Details," page 48 section of this document for a listing of response reason codes.</p>
4	Response Reason Text	<p>Value: A brief description of the result, which corresponds with the response reason code</p> <p>Format: Text</p> <p>Notes: You can generally use this text to display a transaction result or error to the customer. However, review the "Response Code Details," page 48 section of this document to identify any specific texts you do not want to pass to the customer.</p>
5	Authorization Code	<p>Value: The authorization or approval code</p> <p>Format: 6 characters</p>

Table 16 Payment Gateway Response Fields (Continued)

Order	Field Name	Description
6	AVS Response	<p>Value: The Address Verification Service (AVS) response code</p> <p>Format:</p> <ul style="list-style-type: none"> ■ A = Address (Street) matches, ZIP does not ■ B = Address information not provided for AVS check ■ E = AVS error ■ G = Non-U.S. Card Issuing Bank ■ N = No Match on Address (Street) or ZIP ■ P = AVS not applicable for this transaction ■ R = Retry—System unavailable or timed out ■ S = Service not supported by issuer ■ U = Address information is unavailable ■ W = Nine digit ZIP matches, Address (Street) does not ■ X = Address (Street) and nine digit ZIP match ■ Y = Address (Street) and five digit ZIP match ■ Z = Five digit ZIP matches, Address (Street) does not <p>Notes: Indicates the result of the AVS filter.</p> <p>For more information about AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
7	Transaction ID	<p>Value: The payment gateway-assigned identification number for the transaction</p> <p>Format: When <code>x_test_request</code> is set to a positive response, or when Test Mode is enabled on the payment gateway, this value will be "0."</p> <p>Notes: This value must be used for any follow on transactions such as a CREDIT, PRIOR_AUTH_CAPTURE or VOID.</p>
8	Invoice Number	<p>Value: The merchant-assigned invoice number for the transaction</p> <p>Format: Up to 20 characters (no symbols)</p>
9	Description	<p>Value: The transaction description</p> <p>Format: Up to 255 characters (no symbols)</p>
10	Amount	<p>Value: The amount of the transaction</p> <p>Format: Up to 15 digits</p>
11	Method	<p>Value: The payment method</p> <p>CC or ECHECK</p>
12	Transaction Type	<p>Value: The type of credit card transaction</p> <p>Format: AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT, PRIOR_AUTH_CAPTUREVOID</p>

Table 16 Payment Gateway Response Fields (Continued)

Order	Field Name	Description
13	Customer ID	Value: The merchant-assigned customer ID Format: Up to 20 characters (no symbols)
14	First Name	Value: The first name associated with the customer's billing address Format: Up to 50 characters (no symbols)
15	Last Name	Value: The last name associated with the customer's billing address Format: Up to 50 characters (no symbols)
16	Company	Value: The company associated with the customer's billing address Format: Up to 50 characters (no symbols)
17	Address	Value: The customer's billing address Format: Up to 60 characters (no symbols)
18	City	Value: The city of the customer's billing address Format: Up to 40 characters (no symbols)
19	State	Value: The state of the customer's billing address Format: Up to 40 characters (no symbols) or a valid two-character state code
20	ZIP Code	Value: The ZIP code of the customer's billing address Format: Up to 20 characters (no symbols)
21	Country	Value: The country of the customer's billing address Format: Up to 60 characters (no symbols)
22	Phone	Value: The phone number associated with the customer's billing address Format: Up to 25 digits (no letters). For example, (123)123-1234
23	Fax	Value: The fax number associated with the customer's billing address Format: Up to 25 digits (no letters). For example, (123)123-1234
24	Email Address	Value: The customer's valid email address Format: Up to 255 characters
25	Ship To First Name	Value: The first name associated with the customer's shipping address Format: Up to 50 characters (no symbols)
26	Ship To Last Name	Value: The last name associated with the customer's shipping address Format: Up to 50 characters (no symbols)

Table 16 Payment Gateway Response Fields (Continued)

Order	Field Name	Description
27	Ship To Company	Value: The company associated with the customer's shipping address Format: Up to 50 characters (no symbols)
28	Ship To Address	Value: The customer's shipping address Format: Up to 60 characters (no symbols)
29	Ship To City	Value: The city of the customer's shipping address Format: Up to 40 characters (no symbols)
30	Ship To State	Value: The state of the customer's shipping address Format: Up to 40 characters (no symbols) or a valid two-character state code
31	Ship To ZIP Code	Value: The ZIP code of the customer's shipping address Format: Up to 20 characters (no symbols)
32	Ship To Country	Value: The country of the customer's shipping address Format: Up to 60 characters (no symbols)
33	Tax	Value: The tax amount charged Format: Numeric Notes: Delimited tax information is not included in the transaction response.
34	Duty	Value: The duty amount charged Format: Numeric Notes: Delimited duty information is not included in the transaction response.
35	Freight	Value: The freight amount charged Format: Numeric Notes: Delimited freight information is not included in the transaction response.
36	Tax Exempt	Value: The tax exempt status Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0
37	Purchase Order Number	Value: The merchant assigned purchase order number Format: Up to 25 characters (no symbols)
38	MD5 Hash	Value: The payment gateway-generated MD5 hash value that can be used to authenticate the transaction response. Notes: Optional. Transaction responses are returned using SSL/TLS, so this field is useful mainly as a redundant security check.

Table 16 Payment Gateway Response Fields (Continued)

Order	Field Name	Description
39	Card Code Response	<p>Value: The card code verification (CCV) response code</p> <p>Format:</p> <ul style="list-style-type: none"> ■ M = Match ■ N = No Match ■ P = Not Processed ■ S = Should have been present ■ U = Issuer unable to process request <p>Notes: Indicates the result of the CCV filter.</p> <p>For more information about CCV, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
40	Cardholder Authentication Verification Response	<p>Value: The cardholder authentication verification response code</p> <p>Format: Blank or not present = CAVV not validated</p> <ul style="list-style-type: none"> ■ 0 = CAVV not validated because erroneous data was submitted ■ 1 = CAVV failed validation ■ 2 = CAVV passed validation ■ 3 = CAVV validation could not be performed; issuer attempt incomplete ■ 4 = CAVV validation could not be performed; issuer system error ■ 5 = Reserved for future use ■ 6 = Reserved for future use ■ 7 = CAVV attempt – failed validation – issuer available (U.S.-issued card/non-U.S acquirer) ■ 8 = CAVV attempt – passed validation – issuer available (U.S.-issued card/non-U.S. acquirer) ■ 9 = CAVV attempt – failed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer) ■ A = CAVV attempt – passed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer) ■ B = CAVV passed validation, information only, no liability shift
51	Account Number	<p>Value: Last 4 digits of the card provided</p> <p>Format: Alphanumeric (XXXX6835)</p> <p>Notes: This field is returned with all transactions.</p>
52	Card Type	<p>Value: Visa, MasterCard, American Express, Discover, Diners Club, JCB</p> <p>Format: Text</p>

Table 16 Payment Gateway Response Fields (Continued)

Order	Field Name	Description
53	Split Tender ID	<p>Value: 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</p> <p>Format: Alphanumeric</p> <p>Notes: This is only returned in the reply message for the first transaction that receives a partial authorization.</p>
54	Requested Amount	<p>Value: Amount requested in the original authorization</p> <p>Format: Numeric</p> <p>Notes: This is present if the current transaction is for a prepaid card or if a splitTenderId was sent in.</p>
55	Balance On Card	<p>Value: Balance on the debit card or prepaid card</p> <p>Format: Numeric</p> <p>Notes: Can be a positive or negative number. This has a value only if the current transaction is for a prepaid card</p>

Response for Duplicate Transactions

You can specify the window of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction (based on credit card number, invoice number, amount, billing address information, transaction type, etc.) using the duplicate window field (*x_duplicate_window*). The value for this field can range from 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 default window of 2 minutes, the payment gateway response will contain the response code of 3 (processing error) with a response reason code of 11 (duplicate transaction) with no additional details.

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 payment gateway response for the duplicate transaction will include the response code and response reason code listed above, as well as information about the original transaction (as outlined below).

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

- AVS result
- CCV result
- Transaction ID

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 are processed normally, whether specified in the transaction request or after the payment gateway's default 2 minute duplicate window.

AIM Transaction Response Types

There are two versions of the AIM response string, version 3.0 and version 3.1.

Version 3.0

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

Example

The following is a sample transaction query string:

```
https://test.authorize.net/gateway/transact.dll?x_
login=YourAPILogin&x_tran_key=YourTransactionKey&x_delim_data=true&x_
relay_response=false&x_card_num=4111111111111111&x_exp_date=1010&x_
amount=1.00&merchant_defined_field1=merchant-defined-field_1&merchant_
defined_field2=merchant_defined_field_2
```

The response would be:

```
1,1,1,This transaction has been
approved.,iKUUAm,Y,2149207036,,,1.00,CC,auth_
capture,,,,,,,,,,,,,,,,,,,,,433AF62576BFC33D0B7B8A75FB229220,merc
hant defined field 1,merchant defined field 2
```

Version 3.1

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

Example

The following is a sample transaction query string:

```
https://test.authorize.net/gateway/transact.dll?x_
login=99W58L5veksj&x_tran_key=2jL4g9447PQJd3uF&x_delim_data=true&x_
relay_response=false&x_card_num=4111111111111111&x_exp_date=1010&x_
```

```
amount=1.00&merchant_defined_field1=merchant defined field 1&merchant_
defined_field2=merchant defined field 2
```

The response would be:

```
1,1,1,This transaction has been
approved.,tt9ieF,Y,2149207083,,,1.00,CC,auth_
capture,,,,,,,,,,,,,,,,,,,,DE9F3FAD313D32144FA7EDF4B27CCE02,,2,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,merchant defined field 1,merchant defined
field 2
```

Setting the Transaction Version

To set the transaction version, do the following:

- Step 1** Log on to the Merchant Interface
- Step 2** Select **Settings** from the Main Menu
- Step 3** Click **Transaction Version** in the Transaction Response section
- Step 4** Change the Transaction Version by using the drop-down box
- Step 5** Click **Submit** to save changes

You can also set the Transaction version on a per-transaction basis by using `x_version`.



Note

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

Response Code Details

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

- **Response Code** indicates the overall status of the transaction with possible values of approved, declined, errored, or held for review.
- **Response Reason Code** is a numeric representation of a more specific reason for the transaction status.

- **Response Reason Text** details the specific reason for the transaction status. This information can be returned to the merchant and/or customer to provide more information about the status of the transaction.

Table 17 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.

Table 18 Response Reason Code Text

Response Code	Response Reason Code	Response Reason Text	Notes
1	1	This transaction has been approved.	
2	2	This transaction has been declined.	
2	3	This transaction has been declined.	
2	4	This transaction has been declined.	The code returned from the processor indicating that the card used needs to be picked up.
3	5	A valid amount is required.	The value submitted in the amount field did not pass validation for a number.
3	6	The credit card number is invalid.	
3	7	The credit card expiration date is invalid.	The format of the date submitted was incorrect.
3	8	The credit card has expired.	
3	9	The ABA code is invalid.	The value submitted in the x_bank_aba_code field did not pass validation or was not for a valid financial institution.
3	10	The account number is invalid.	The value submitted in the x_bank_acct_num field did not pass validation.
3	11	A duplicate transaction has been submitted.	A transaction with identical amount 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.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	13	The merchant API Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applicable only to SIM and WebLink APIs.
3	15	The transaction ID is invalid.	The transaction ID value is non-numeric or was not present for a transaction that requires it (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 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	ACH transactions are not accepted by this merchant.	The merchant does not accept electronic checks.
3	19 - 23	An error occurred during processing. Please try again in 5 minutes.	
3	24	The Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
3	25 - 26	An error occurred during processing. Please try again in 5 minutes.	
2	27	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	
2	28	The merchant does not accept this type of credit card.	The Merchant ID at the processor was not configured to accept this card type.
2	29	The Paymentech identification numbers are incorrect. Call Merchant Service Provider.	
2	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
2	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	32	This reason code is reserved or not applicable to this API.	

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	33	<i>FIELD</i> cannot be left blank.	The word <i>FIELD</i> will be replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in. See the Form Fields section of the Merchant Integration Guide for details.
2	34	The VITAL identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	35	An error occurred during processing. Call Merchant Service 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.	
2	38	The Global Payment System identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	40	This transaction must be encrypted.	
2	41	This transaction has been declined.	Only merchants set up for the FraudScreen.Net service would receive this decline. This code will be returned if a given transaction's fraud score is higher than the threshold set by the merchant.
3	43	The merchant was incorrectly set up at the processor. Call your Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	44	This transaction has been declined.	The card code submitted with the transaction did not match the card code on file at the card issuing bank and the transaction was declined.
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	This occurs if the merchant tries to capture funds greater than the amount of the original authorization-only transaction.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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	The transaction type was invalid for ACH transactions.	If x_method = ECHECK, x_type cannot be set to CAPTURE_ONLY.
3	54	The referenced transaction does not meet the criteria for issuing a credit.	
3	55	The sum of credits against the referenced transaction would exceed the original debit amount.	The transaction is rejected if the sum of this credit and prior credits exceeds the original debit amount
3	56	This merchant accepts ACH transactions only; no credit card transactions are accepted.	The merchant processes eCheck.Net transactions only and does not accept credit cards.
3	57 - 63	An error occurred in processing. Please try again in 5 minutes.	
2	65	This transaction has been declined.	The transaction was declined because the merchant configured their account through the Merchant Interface to reject transactions with certain values for a Card Code mismatch.
3	66	This transaction cannot be accepted for processing.	The transaction did not meet gateway security guidelines.
3	68	The version parameter is invalid.	The value submitted in x_version was invalid.
3	69	The transaction type is invalid.	The value submitted in x_type was invalid.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	70	The transaction method is invalid.	The value submitted in x_method was invalid.
3	71	The bank account type is invalid.	The value submitted in x_bank_acct_type was invalid.
3	72	The authorization code is invalid.	The value submitted in x_auth_code was more than six characters in length.
3	73	The driver's license date of birth is invalid.	The format of the value submitted in x_drivers_license_dob was invalid.
3	74	The duty amount is invalid.	The value submitted in x_duty failed format validation.
3	75	The freight amount is invalid.	The value submitted in x_freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in x_tax failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in x_customer_tax_id failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in x_card_code failed format validation.
3	79	The driver's license number is invalid.	The value submitted in x_drivers_license_num failed format validation.
3	80	The driver's license state is invalid.	The value submitted in x_drivers_license_state failed format validation.
3	81	The requested form type is invalid.	The merchant requested an integration method not compatible with the AIM API.
3	82	Scripts are only supported in version 2.5.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	83	The requested script is either invalid or no longer supported.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	84	This reason code is reserved or not applicable to this API.	
3	85	This reason code is reserved or not applicable to this API.	
3	86	This reason code is reserved or not applicable to this API.	
3	87	This reason code is reserved or not applicable to this API.	

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	88	This reason code is reserved or not applicable to this API.	
3	89	This reason code is reserved or not applicable to this API.	
3	90	This reason code is reserved or not applicable to this API.	
3	91	Version 2.5 is no longer supported.	
3	92	The gateway no longer supports the requested method of integration.	
3	97	This transaction cannot be accepted.	Applicable only to SIM API. Fingerprints are valid only for a short period of time. If the fingerprint is more than one hour old or more than 15 minutes into the future, it will be rejected. This code indicates that the transaction fingerprint has expired.
3	98	This transaction cannot be accepted.	Applicable only to SIM API. The transaction fingerprint has already been used.
3	99	This transaction cannot be accepted.	Applicable only to SIM API. The server-generated fingerprint does not match the merchant-specified fingerprint in the x_fp_hash field.
3	100	The eCheck.Net type is invalid.	Applicable only to eCheck.Net. The value specified in the x_echeck_type field is invalid.
3	101	The given name on the account and/or the account type does not match the actual account.	Applicable only to eCheck.Net. The specified name on the account and/or the account type do not match the NOC record for this account.
3	102	This request cannot be accepted.	A password or Transaction Key was submitted with this WebLink request. This is a high security risk.
3	103	This transaction cannot be accepted.	A valid fingerprint, Transaction Key, or password is required for this transaction.
3	104	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for country failed validation.
3	105	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for city and country failed validation.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	106	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for company failed validation.
3	107	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name does not contain valid characters.
3	116	The authentication indicator is invalid.	This error applies only to Verified by Visa and MasterCard SecureCode transactions. The ECI value for a Visa transaction; or the UCAF indicator for a MasterCard transaction submitted in the x_authentication_indicator field is invalid.
3	117	The cardholder authentication value is invalid.	This error applies only to Verified by Visa and MasterCard SecureCode transactions. The CAVV for a Visa transaction; or the AVV/UCAF for a MasterCard transaction is invalid.
3	118	The combination of authentication indicator and cardholder authentication value is invalid.	This error applies only to Verified by Visa and MasterCard SecureCode transactions. The combination of authentication indicator and cardholder authentication value for a Visa or MasterCard transaction is invalid. For more information, see the " Cardholder Authentication ," page 35" section of this document.
3	119	Transactions having cardholder authentication values cannot be marked as recurring.	This error applies only to Verified by Visa and MasterCard SecureCode transactions. Transactions submitted with a value in x_authentication_indicator <i>and</i> x_recurring_billing=YES will be rejected.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	120	An error occurred during processing. Please try again.	The system-generated void for the original timed-out transaction failed. (The original transaction timed out while waiting for a response from the authorizer.)
3	121	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a database error.)
3	122	An error occurred during 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.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.
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. Debit the credit card instead of voiding the credit.
3	180	An error occurred during processing. Please try again.	The processor response format is invalid.
3	181	An error occurred during processing. Please try again.	The system-generated void for the original invalid transaction failed. (The original transaction included an invalid processor response format.)
3	185	This reason code is reserved or not applicable to this API.	
4	193	The transaction is currently under review.	The transaction was placed under review by the risk management system.
2	200	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The credit card number is invalid.
2	201	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The expiration date is invalid.
2	202	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The transaction type is invalid.
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.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	205	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The value submitted in the merchant number field is invalid.
2	206	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	207	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant account is closed.
2	208	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant is not on file.
2	209	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Communication with the processor could not be established.
2	210	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant type is incorrect.
2	211	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The cardholder is not on file.
2	212	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The bank configuration is not on file
2	213	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant assessment code is incorrect.
2	214	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This 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.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	218	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The PIN block format or PIN availability value is invalid.
2	219	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ETC void is unmatched.
2	220	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The primary CPU is not available.
2	221	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The SE number is invalid.
2	222	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Duplicate auth request (from INAS).
2	223	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced an unspecified error.
2	224	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Re-enter the transaction.
3	243	Recurring billing is not allowed for this eCheck.Net type.	The combination of values submitted for x_recurring_billing and x_echeck_type is not allowed.
3	244	This eCheck.Net type is not allowed for this Bank Account Type.	The combination of values submitted for x_bank_acct_type and x_echeck_type is not allowed.
3	245	This eCheck.Net type is not allowed when using the payment gateway hosted payment form.	The value submitted for x_echeck_type is not allowed when using the payment gateway hosted payment form.
3	246	This eCheck.Net type is not allowed.	The merchant's payment gateway account is not enabled to submit the eCheck.Net type.
3	247	This eCheck.Net type is not allowed.	The combination of values submitted for x_type and x_echeck_type is not allowed.
3	248	The check number is invalid.	Invalid check number. Check number is limited to 15 alphanumeric characters.

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	250	This transaction has been declined.	This transaction was submitted from a blocked IP address.
2	251	This transaction has been declined.	The transaction was declined as a result of triggering a Fraud Detection Suite filter.
4	252	Your order has been received. Thank you for your business!	The transaction was accepted, but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
4	253	Your order has been received. Thank you for your business!	The transaction was accepted and was authorized, but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
2	254	Your transaction has been declined.	The transaction was declined after manual review.
3	261	An error occurred during processing. Please try again.	The transaction experienced an error during sensitive data encryption and was not processed. Try again.
3	270	The line item [item number] is invalid.	A value submitted in x_line_item for the item referenced is invalid.
3	271	The number of line items submitted is not allowed. A maximum of 30 line items can be submitted.	The number of line items submitted exceeds the allowed maximum of 30.
3	288	Merchant is not registered as a Cardholder Authentication participant. This transaction cannot be accepted.	The merchant has not indicated participation in any Cardholder Authentication Programs in the Merchant Interface.
3	289	This processor does not accept zero dollar authorization for this card type.	Your credit card processing service does not yet accept zero dollar authorizations for Visa credit cards. You can find your credit card processor listed on your merchant profile.
3	290	One or more required AVS values for zero dollar authorization were not submitted.	When submitting authorization requests for Visa, you must enter the address and zip code fields.
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.	The merchant must have partial authorization enabled in the Merchant Interface in order to receive this error.
3	296	The specified SplitTenderId is not valid.	

Table 18 Response Reason Code Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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 invalid.
3	301	The device batch ID is invalid.	The value submitted for x_device_batch_id is invalid.
3	302	The reversal flag is invalid.	The value submitted for x_reversal is invalid.
3	303	The device batch is full. Please close the batch.	The current device batch must be closed manually from the POS device.
3	304	The original transaction is in a closed batch.	The original transaction has been settled and cannot be reversed.
3	305	The merchant is configured for auto-close.	This merchant is configured for auto-close and cannot manually close batches.
3	306	The batch is already closed.	The batch is already closed.
1	307	The reversal was processed successfully.	The reversal was processed successfully.
1	308	Original transaction for reversal not found.	The transaction submitted for reversal was not found.
3	309	The device has been disabled.	The device has been disabled.
1	310	This transaction has already been voided.	This transaction has already been voided.
1	311	This transaction has already been 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**

A very helpful tool for troubleshooting Reason Response Codes is available in our Integration Center at <http://developer.authorize.net/tools/responsereasoncode>.

Email Receipt

Merchants can choose to send a payment gateway-generated email receipt to customers who provide an email address with their transaction. The email receipt includes a summary and results of the transaction. To the customer, this email appears to be sent from the merchant contact that is configured as the Email Sender in the Merchant Interface.

To send the payment gateway-generated customer email receipt, submit the API fields in the following table, with the transaction request string. These settings can also be configured in the Merchant Interface.

For more information about configuring these settings, see the *Merchant Integration Guide* at <http://www.authorize.net/support/merchant/>.

Fields are name/value pairs with the syntax of:

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

Table 19 Email Receipt Fields

Field Name	Description
x_email	<p>Value: The customer's valid email address</p> <p>Format: Up to 255 characters. For example, janedoe@customer.com</p> <p>Notes: The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p> <p>For more information about Email Receipts, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_email_customer	<p>Value: The customer email receipt status</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether an email receipt should be sent to the customer.</p> <p>If set to TRUE, the payment gateway will send an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer.</p> <p>If no value is submitted, the payment gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has enabled the setting. If this field is not submitted and the setting is disabled in the Merchant Interface, no email is sent.</p> <p>For more information about configuring Email Receipts in the Merchant Interface, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_header_email_receipt	<p>Value: The email receipt header</p> <p>Format: Plain text</p> <p>Notes: This text will appear as the header of the email receipt sent to the customer.</p>

Table 19 Email Receipt Fields (Continued)

Field Name	Description
x_footer_email_receipt	<p>Value: The email receipt footer</p> <p>Format: Plain text</p> <p>Notes: This text will appear as the footer on the email receipt sent to the customer.</p>
x_merchant_email	<p>Value: Any valid email address</p> <p>Format: Up to 255 characters</p> <p>Only one email address per variable is allowed</p> <p>Notes: Email address to which the merchant's copy of the customer confirmation email should be sent. If a value is submitted, an email will be sent to this address as well as the address(es) configured in the Merchant Interface.</p> <p>Warning If it is included, it can subject the merchant to spam on their business email address, because it announces where the receipt gets returned to, and gives a hint where relay response or silent post information may be sent.</p>

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

Test Transactions

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

Ideally, you should test your integration in the following phases:

Phase 1: Testing in the test environment

In this environment, test transactions are posted to **<https://test.authorize.net/gateway/transact.dll>**. Although this is a staging environment, its behavior mimics the live payment gateway. Transactions submitted to the test environment using a developer test account are not submitted to financial institutions for authorization and are not stored in the Merchant Interface.

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



Note

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

Phase 2: Testing in the live environment

After you successfully test the integration in the developer test environment, you can plug the merchant's Authorize.Net Payment Gateway API Login ID and Transaction Key into the integration for testing against the live environment. Developer test account credentials are not accepted by the live payment gateway.

In this phase, you can test the integration in one of two ways:

- a** By including the `x_test_request` field with a value of "TRUE" in the transaction request string to **<https://secure.authorize.net/gateway/transact.dll>**. See the sample below.

Example Submitting the test request field

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

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

**Note**

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

Phase 3

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

**Note**

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

Testing to Generate Specific Transaction Results

When testing transaction results in the developer test environment as well as the production environment, you can produce a specific response reason code by submitting a test transaction that uses a test credit card number designed to generate specific transaction results: this is Visa test credit card number "422222222222." This card number is intended for testing and should only be used for that purpose. Submit the test transaction either by placing the account in Test Mode, or submitting `x_test_`

request=TRUE, with a dollar amount equal to the response reason code you would like to produce.

For example, to test the AVS response reason code number 27, submit the test transaction with the credit card number "422222222222" and the amount "27.00."

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

For more information about response reason codes, see [Chapter 4, "Transaction Response,"](#) on page 39 of this guide.

Fields by Transaction Type

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

- The minimum fields required to submit a transaction
- Additional fields that are required in order to configure advanced features of AIM
- “Best practice” fields, or fields that the payment gateway recommends should be submitted on a per-transaction basis in order to maintain a strong connection to the payment gateway—for example, to prevent possible conflicts if integration settings in the Merchant Interface are inadvertently changed.

Minimum Required Fields

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

Table 20 Minimum Required Fields

	Authorization and Capture	Authorization	Prior Authorization and Capture	Capture Only	Credit	Void
Merchant Information	x_login	x_login	x_login	x_login	x_login	x_login
	x_tran_key	x_tran_key	x_tran_key	x_tran_key	x_tran_key	x_tran_key
Transaction Information	x_type = AUTH_CAPTURE	x_type = AUTH_ONLY	x_type = PRIOR_AUTH_CAPTURE	x_type = CAPTURE_ONLY	x_type = CREDIT	x_type = VOID
			x_trans_id or x_split_tender_id	x_auth_code	x_trans_id	x_trans_id or x_split_tender_id
Payment Information	x_amount	x_amount	x_amount	x_amount	x_amount	N/A
	x_card_num	x_card_num	(required only	x_card_num	x_card_num	
	x_exp_date	x_exp_date	when less than the original authorization amount)	x_exp_date	x_exp_date**	

* For merchants registered for expanded credit capabilities (ECC), a Transaction ID should NOT be submitted for Unlinked Credits. For more information, see the "[Credit Card Transaction Types](#)," page 16 section of this document.

** The expiration date is only required for Unlinked Credits.

Required Fields for Additional AIM Features

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

Table 21 Required Fields for Additional AIM Features

	Authorization and Capture	Authorization Only	Prior Authorization and Capture	Capture Only	Credit	Void
Itemized Order Information	x_line_item	x_line_item	x_line_item	x_line_item	x_line_item	N/A
	x_tax†	x_tax†	x_tax†	x_tax†	x_tax†	
	x_freight†	x_freight†	x_freight†	x_freight†	x_freight†	
	x_duty†	x_duty†	x_duty†	x_duty†	x_duty†	
Cardholder Authentication	x_authentication_indicator	x_authentication_indicator	N/A	N/A	N/A	N/A
	x_cardholder_authentication_value	x_cardholder_authentication_value				
Advanced Fraud Detection Suite™ (AFDS)	x_customer_ip (required only when the merchant is using customer-IP based AFDS filters)	x_customer_ip (required only when the merchant is using customer-IP based AFDS filters)	N/A	N/A	N/A	N/A
eCheck.Net	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE
	x_relay_response = FALSE	x_relay_response = FALSE	x_relay_response = FALSE	x_relay_response = FALSE	x_relay_response = FALSE	x_relay_response = FALSE

† These fields can support either a straight numeric value, or line item details similar to x_line_item.



Note

For Prior Authorization and 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 applies.

Best Practice Fields

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

Table 22 Best Practice Fields

	Authorization and Capture	Authorization Only	Prior Authorization and Capture	Capture Only	Credit	Void
Transaction Information	x_version = 3.1	x_version = 3.1	x_version = 3.1	x_version = 3.1	x_version = 3.1	x_version = 3.1
Transaction Response	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE	x_delim_data = TRUE
	x_delim_char	x_delim_char	x_delim_char	x_delim_char	x_delim_char	x_delim_char
	x_encap_char	x_encap_char	x_encap_char	x_encap_char	x_encap_char	x_encap_char
	x_relay_response = FALSE*	x_relay_response = FALSE*	x_relay_response = FALSE*	x_relay_response = FALSE*	x_relay_response = FALSE*	x_relay_response = FALSE*

* The *x_relay_response* field is not technically an AIM feature; however, it is recommended that you submit this field on a per-transaction basis with the value of "FALSE" as a best practice to further define the AIM transaction format.

Alphabetized List of API Fields

Table 23 Alphabetized List of API Fields

Field Name	Description
x_address	<p>Required only when using a European Payment Processor.</p> <p>Value: The customer's billing address</p> <p>Format: Up to 60 characters (no symbols)</p> <p>Notes: Required if the merchant would like to use the Address Verification Service security feature.</p> <p>For more information on AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p> <p>Required for Zero Dollar Authorizations for Visa verification transactions.</p>
x_allow_partial_Auth	<p>Optional</p> <p>Value: True, False</p> <p>Format: True, False, T, F</p> <p>Notes: Set this value if the merchant would like to override a setting in the Merchant Interface.</p>
x_amount	<p>Required if x_type = AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT</p> <p>Value: The amount of the transaction</p> <p>Format: Up to 15 digits with a decimal point (no dollar symbol). for example, 8.95</p> <p>Notes: The total amount to be charged or credited <i>including</i> tax, shipping and any other charges. The amount can either be hard coded or posted to a script.</p>
x_auth_code	<p>Required if x_type = CAPTURE_ONLY</p> <p>Value: The authorization code for an original transaction not authorized on the payment gateway</p> <p>Format: 6 characters</p> <p>Notes: Required only for CAPTURE_ONLY transactions. See the "Credit Card Transaction Types," page 16" section of this document.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_authentication_ indicator	<p>Optional</p> <p>Value: The electronic commerce indicator (ECI) value for a Visa transaction; or the universal cardholder authentication field indicator (UCAF) for a MasterCard transaction obtained by the merchant after the authentication process.</p> <p>Format: Special characters included in this value must be URL encoded.</p> <p>Notes: Required only for AUTH_ONLY and AUTH_CAPTURE transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored.</p> <p>This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.</p>
x_card_code	<p>Optional</p> <p>Value: The customer's card code</p> <p>Format: Must be a valid CVV2, CVC2 or CID value.</p> <p>Notes: The three- or four-digit number on the back of a credit card (on the front for American Express).</p> <p>This field is required if the merchant would like to use the Card Code Verification (CCV) security feature. For more information, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_card_num	<p>Required if x_type = AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT</p> <p>Value: The customer's credit card number</p> <p>When x_type=CREDIT, only the last four digits are required.</p> <p>Format: Between 13 and 16 digits without spaces</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the http://www.authorize.net/files/developerbestpractices.pdf.</p>
x_cardholder_ authentication_value	<p>Optional</p> <p>Value: The cardholder authentication verification value (CAVV) for a Visa transaction; or accountholder authentication value (AVV)/ universal cardholder authentication field (UCAF) for a MasterCard transaction obtained by the merchant after the authentication process.</p> <p>Format: Special characters included in this value must be URL encoded</p> <p>Notes: Required only for AUTH_ONLY and AUTH_CAPTURE transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored.</p> <p>This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.</p>
x_city	<p>Required only when using a European Payment Processor.</p> <p>Value: The city of the customer's billing address</p> <p>Format: Up to 40 characters (no symbols)</p>

Table 23 Alphabetized List of API Fields (Continued)

x_company	Optional Value: The company associated with the customer's billing address Format: Up to 50 characters (no symbols)
x_country	Required only when using a European Payment Processor. Value: The country of the customer's billing address Format: Up to 60 characters (no symbols)
x_currency_code	Optional Value: USD, CAD, or GBP. Format: 3 character string. Notes: If this field is not submitted, the default is the currency selected by the merchant's payment processor. Setting this field to a currency that is not supported by the payment processor will result in an error.
x_cust_id	Optional Value: The merchant-assigned customer ID Format: Up to 20 characters (no symbols) Notes: The unique identifier to represent the customer associated with the transaction. The customer ID must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.
x_customer_ip	Optional Value: The customer's IP address Format: Up to 15 characters (no letters). For example, 255.255.255.255 Notes: The IP address of the customer initiating the transaction. If this value is not passed, it will default to 255.255.255.255. This field is required when using customer-IP based Advanced Fraud Detection Suite™ (AFDS) filters. For more information about AFDS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ .

Table 23 Alphabetized List of API Fields (Continued)

x_delim_char	<p>Optional</p> <p>Value: The delimiting character; for example:</p> <ul style="list-style-type: none"> , (comma) (pipe) “ (double quotes) ‘ (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star) <p>Notes: The character that is used to separate fields in the transaction response. The payment gateway will use the character passed in this field or the value stored in the Merchant Interface if no value is passed.</p> <p>If this field is passed and the value is null, it overrides the value stored in the Merchant Interface and the transaction response contains no delimiting character.</p> <p>It is recommended that you submit this field on a per-transaction basis to be sure that transaction responses are returned in the correct format.</p>
x_delim_data	<p>Required for AIM transactions</p> <p>Value: The request to receive a delimited transaction response</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: In order to receive a delimited response from the payment gateway, this field must be submitted with a value of TRUE or the merchant has to configure a delimited response through the Merchant Interface.</p> <p>It is recommended that you submit this field on a per-transaction basis to be sure that transaction responses are returned in the correct format.</p> <p>This field is paired with x_relay_response. If one is set to True, the other must be set to False.</p>
x_description	<p>Optional</p> <p>Value: The transaction description</p> <p>Format: Up to 255 (no symbols)</p> <p>Notes: The description must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_duplicate_window	<p>Optional</p> <p>Value: The window of time after the submission of a transaction that a duplicate transaction can not be submitted</p> <p>Format: Any value between 0 and 28800 (no commas)</p> <p>Notes: 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).</p> <p>If a value less than 0 is sent, the payment gateway will default to 0 seconds. If a value greater than 28800 is sent, the payment gateway will default to 28800. If no value is sent, the payment gateway will default to 2 minutes (120 seconds).</p> <p>If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See the "Response for Duplicate Transactions," page 46 section of this guide for more information.</p>
x_duty	<p>Optional</p> <p>Value: The valid duty amount OR delimited duty information</p> <p>Format: When submitting delimited duty information, values must be delimited by a bracketed pipe < ></p> <p>Notes: The duty amount charged OR when submitting this information using the transaction request, delimited duty information including the duty name, description, and amount is also allowed. The total amount of the transaction in x_amount must include this amount.</p> <p>Delimited duty information elements include:</p> <ul style="list-style-type: none"> ■ Duty item name< > ■ Duty description< > ■ Duty amount: The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must <i>include</i> this amount. <p>Example: x_duty=Duty1< >export< >15.00&</p>
x_email	<p>Required only when using a European Payment Processor.</p> <p>Value: The customer's valid email address</p> <p>Format: Up to 255 characters. For example, janedoe@customer.com</p> <p>Notes: The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p> <p>For more information about Email Receipts, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_email_customer	<p>Optional</p> <p>Value: The customer email receipt status</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether an email receipt should be sent to the customer.</p> <p>If set to TRUE, the payment gateway will send an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer.</p> <p>If no value is submitted, the payment gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has enabled the setting. If this field is not submitted and the setting is disabled in the Merchant Interface, no email is sent.</p> <p>For more information about configuring Email Receipts in the Merchant Interface, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_encap_char	<p>Optional</p> <p>Value: The encapsulating character</p> <p> (pipe) " (double quotes) ' (single quote) : (colon) ; (semicolon) / (forward slash) \ (back slash) - (dash) * (star)</p> <p>Notes: The character that is used to encapsulate the fields in the transaction response. This is only necessary if it is possible that your delimiting character could be included in any field values.</p> <p>The payment gateway will use the character passed in this field or the value stored in the Merchant Interface if no value is passed.</p>
x_exp_date	<p>Required</p> <p>Value: The customer's credit card expiration date</p> <p>Format: MMY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY</p> <p>Notes: This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard.</p> <p>For more information about PCI, see the <i>Developer Security Best Practices White Paper</i> at http://www.authorize.net/files/developerbestpractices.pdf.</p>
x_fax	<p>Optional</p> <p>Value: The fax number associated with the customer's billing address</p> <p>Format: Up to 25 digits (no letters). For example, (123)123-1234.</p>
x_first_name	<p>Required only when using a European Payment Processor.</p> <p>Value: The first name associated with the customer's billing address</p> <p>Format: Up to 50 characters (no symbols)</p>

Table 23 Alphabetized List of API Fields (Continued)

x_footer_email_receipt	<p>Optional</p> <p>Value: The email receipt footer</p> <p>Format: Plain text</p> <p>Notes: This text will appear as the footer on the email receipt sent to the customer.</p>
x_freight	<p>Optional</p> <p>Value: The valid freight amount OR delimited freight information</p> <p>Format: When submitting delimited freight information, values must be delimited by a bracketed pipe < ></p> <p>Notes: The freight amount charged OR when submitting this information using the transaction request string, delimited freight information including the freight name, description, and amount is also allowed. The total amount of the transaction in x_amount must include this amount.</p> <p>Delimited freight information elements include:</p> <ul style="list-style-type: none"> ■ Freight item name< > ■ Freight description< > ■ Freight amount: The freight item amount. The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must include this amount. <p>Example: x_freight=Freight< >ground overnight< >12.95&</p>
x_header_email_receipt	<p>Optional</p> <p>Value: The email receipt header</p> <p>Format: Plain text</p> <p>Notes: This text will appear as the header of the email receipt sent to the customer.</p>
x_invoice_num	<p>Optional</p> <p>Value: The merchant assigned invoice number for the transaction</p> <p>Format: Up to 20 characters (no symbols)</p> <p>Notes: The invoice number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>
x_last_name	<p>Required only when using a European Payment Processor.</p> <p>Value: The last name associated with the customer's billing address</p> <p>Format: Up to 50 characters (no symbols)</p>

Table 23 Alphabetized List of API Fields (Continued)

x_line_item	<p>Optional</p> <p>All line item elements are required when this field is submitted. Up to 30 may be submitted.</p> <p>Value: Any string</p> <p>Format: Line item values must be delimited by a bracketed pipe < >, and must be listed in the order shown below.</p> <p>Notes: Itemized order information.</p> <p>Delimited item information elements include:</p> <ul style="list-style-type: none"> ■ Item ID< >: Up to 31 characters ■ Item name< >: Up to 31 characters ■ Item description< >: Up to 255 characters ■ Item quantity< >: Up to two decimal places. ■ Item price (per unit)< >: Up to two decimal places. Must be a positive number. The dollar sign (\$) is not allowed when submitting delimited information. Excludes tax, freight, and duty. ■ Item taxable: Values can be TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0 <p>Example: x_line_item=item1< >golf balls< >< >2< >18.95< >Y</p>
x_login	<p>Required</p> <p>Value: The merchant's unique API Login ID</p> <p>Format: Up to 20 characters</p> <p>Notes: The merchant API Login ID is provided in the Merchant Interface.</p> <p>The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/ for more information.</p>
x_market_type	<p>Optional</p> <p>Value: One of the following.</p> <p>0 for e-commerce</p> <p>1 for moto</p> <p>2 for retail.</p> <p>Notes: If your account type is Card Present, the default is 2 and only 2 can be used. If your account type is blended, the default is 0, but x_market_type can be overridden.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_merchant_email	<p>Optional</p> <p>Value: Any valid email address</p> <p>Format: Up to 255 characters</p> <p>Only one email address per variable is allowed</p> <p>Notes: Email address to which the merchant's copy of the customer confirmation email should be sent. If a value is submitted, an email will be sent to this address as well as the address(es) configured in the Merchant Interface.</p> <p>If included, it can subject the merchant to spam on their business email address, because it announces where the receipt gets returned to, and gives a hint where relay response or silent post information may be sent.</p>
x_method	<p>Optional</p> <p>Value: The payment method</p> <p>Format: CC or ECHECK</p> <p>Notes: The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If left blank, this value will default to CC.</p> <p>For more information about eCheck.Net transaction requirements, see the eCheck.Net Developer Guide.</p>
x_phone	<p>Optional</p> <p>Value: The phone number associated with the customer's billing address</p> <p>Format: Up to 25 digits (no letters). For example, (123)123-1234</p>
x_po_num	<p>Optional</p> <p>Value: The merchant-assigned purchase order number</p> <p>Format: Up to 25 characters (no symbols)</p> <p>Notes: The purchase order number must be created dynamically on the merchant server or provided on a per-transaction basis. The payment gateway does not perform this function.</p>
x_recurring_billing	<p>Optional</p> <p>Value: The recurring billing status</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicating marker used by merchant account providers to identify transactions which originate from merchant hosted recurring billing applications. This value is not affiliated with Automated Recurring Billing.</p>
x_relay_response	<p>Optional</p> <p>Value: The request for a relay response</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: This field, when set to TRUE, instructs the payment gateway to return transaction results to the merchant by means of an HTML form POST to the merchant's Web server for a relay response.</p> <p>Relay response is used for SIM applications. Set this to false if you are using AIM.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_ship_to_address	Optional Value: The customer's shipping address Format: Up to 60 characters (no symbols)
x_ship_to_company	Optional Value: The company associated with the customer's shipping address Format: Up to 50 characters (no symbols)
x_ship_to_country	Optional Value: The country of the customer's shipping address Format: Up to 60 characters (no symbols)
x_ship_to_city	Optional Value: The city of the customer's shipping address Format: Up to 40 characters (no symbols)
x_ship_to_first_name	Optional Value: The first name associated with the customer's shipping address Format: Up to 50 characters (no symbols)
x_ship_to_last_name	Optional Value: The last name associated with the customer's shipping address Format: Up to 50 characters (no symbols)
x_ship_to_state	Optional Value: The state of the customer's shipping address Format: Up to 40 characters (no symbols) or a valid two-character state code
x_ship_to_zip	Optional Value: The ZIP code of the customer's shipping address Format: Up to 20 characters (no symbols)
x_split_tender_id	Optional Value: The payment gateway-assigned ID that links the current authorization request to the original authorization request. Format: Numeric Notes: This value is returned in the reply message from the original authorization request. Transmit this value in subsequent transactions that are related to the same order.
x_state	Required only when using a European Payment Processor. Value: The state of the customer's billing address Format: Up to 40 characters (no symbols) or a valid two-character state code

Table 23 Alphabetized List of API Fields (Continued)

x_tax	<p>Optional</p> <p>Value: The valid tax amount OR the delimited tax information</p> <p>Format: When submitting delimited tax information, values must be delimited by a bracketed pipe < ></p> <p>Notes: The tax amount charged OR when submitting this information using the transaction request string, delimited tax information including the sales tax name, description, and amount is also allowed. The total amount of the transaction in x_amount must include this amount.</p> <p>Delimited tax information elements include:</p> <ul style="list-style-type: none"> ■ Tax item name< > ■ Tax description< > ■ Tax amount: The dollar sign (\$) is not allowed when submitting delimited information. The total amount of the transaction in x_amount must include this amount. <p>Example: x_tax=Tax1< >state tax< >0.09</p>
x_tax_exempt	<p>Optional</p> <p>Value: The tax exempt status</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether the transaction is tax exempt.</p>
x_test_request	<p>Optional</p> <p>Value: The request to process test transactions</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates if the transaction should be processed as a test transaction.</p> <p>See Chapter 5, "Test Transactions," on page 64 of this guide for more information.</p>
x_tran_key	<p>Required for merchant authentication</p> <p>Value: The merchant's unique Transaction Key</p> <p>Format: 16 characters</p> <p>Notes: The merchant Transaction Key is provided in the Merchant Interface and must be stored securely.</p> <p>The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p>
x_trans_id	<p>Required when x_type = CREDIT, PRIOR_AUTH_CAPTURE, VOID</p> <p>Value: The payment gateway assigned transaction ID of the original transaction</p> <p>Format: Numeric</p> <p>Notes: Required only for CREDIT, PRIOR_AUTH_CAPTURE, and VOID transactions.</p> <p>For more information about transaction types, see the "Credit Card Transaction Types," page 16 section of this guide.</p> <p>Do not include this field if you include the x_split_tender_id field.</p>

Table 23 Alphabetized List of API Fields (Continued)

x_type	<p>Optional</p> <p>Value: The type of credit card transaction</p> <p>Format: AUTH_CAPTURE (default), AUTH_ONLY, CAPTURE_ONLY, CREDIT, PRIOR_AUTH_CAPTURE, VOID.</p> <p>Notes: If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted or the value is blank, the payment gateway will process the transaction as an AUTH_CAPTURE.</p>
x_version	<p>Optional, but highly recommended</p> <p>Value: The merchant's transaction version</p> <p>Format: 3.0, 3.1</p> <p>Notes: Indicates to the system the set of fields that will be included in the response: 3.0 is the default version.</p> <p>3.1 allows the merchant to utilize partial authorizations and the Card Code feature, and is the current standard version.</p> <p>It is highly recommended that you submit this field on a per-transaction basis to be sure that the formats of transaction requests and the responses you receive are consistent.</p>
x_zip	<p>Required only when using a European Payment Processor.</p> <p>Value: The ZIP code of the customer's billing address</p> <p>Format: Up to 20 characters (no symbols)</p> <p>Notes: Required if the merchant would like to use the Address Verification Service security feature.</p> <p>For more information on AVS, see the <i>Merchant Integration Guide</i> at http://www.authorize.net/support/merchant/.</p> <p>Required for Zero Dollar Authorizations for Visa verification transactions.</p>