Merchant Web Services API

Advanced Integration Method (AIM)

XML Guide

February 2015

Authorize.Net

Authorize.Net Developer Support
http://developer.authorize.net
Authorize.Net LLC 082007 Ver.2.0

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

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

Authorize.Net Trademarks

Advanced Fraud Detection Suite™

Authorize.Net®

Authorize.Net Your Gateway to IP Transactions™

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

Automated Recurring Billing™

eCheck.Net®

Contents

Revision History

Publish Date	Update
February 2014	Added EVO to the list of payment processors. See "North American Payment Processors," page 12. Added a section about "EVO Billing and Shipping Information," page 42. Added **employeeId** to "createTransactionRequest Elements," page 31.
November 2014	Moved the **profile** block of elements beneath the **payment** block of elements in Table 6, "createTransactionRequest Elements." **payment** elements must precede **profile** elements.
September 2014	Added the **createProfile** element to "createTransactionRequest Elements," page 31. You can use this field to create a customer profile using the billing and shipping information in a transaction request. Added a **profileResponse** section to the end of "createTransactionResponse Fields," page 52. Added the following error messages to "Information and Error Messages," page 87: ■ E00098 ■ E00099 ■ E000100 ■ E000101 ■ E000102
July 2014	Added a **profile** section to the list of API fields in "createTransactionRequest Elements," page 31. You can use this profile section to process payment information using payment and shipping profiles.
June 2014	Updated "Supported Card Readers," page 94 with new contact information for ordering supported encrypting card readers. Added ARC and BOC to echeckType field in "createTransactionRequest Elements," page 31.

Publish Date	Update
February 2014	Added a note about PayPal Express Checkout. See "PayPal Express Checkout," page 15. Updated Developer Support section. See "Developer Support," page 16. Updated default information for the **marketType** field. See "createTransactionRequest Elements," page 31.

CHAPTER 1

Introduction

This guide describes the Web development required to connect an e-commerce Web site, a retail point of sale, 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). This guide provides instructions on submitting transactions using XML; for alternate methods, go to our Developer Center: http://developer.authorize.net/api/aim.

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

- Collecting customer payment information through a custom application
- Generating a receipt to the customer
- Securely transmitting transaction data to the payment gateway for 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 is an ideal integration solution because it allows merchants the highest degree of customization and control over their customers' checkout experience.

About This Guide

This guide provides information that applies to both retail (Card Present) and e-commerce (Card Not Present) implementations. The description of a feature will indicate whether that feature applies to only one implementation. For example, a note might say "This feature is available for retail applications only."

Audience For This Guide

The developers who write the code that integrates merchant websites with the payment gateway are the primary audience for this document. In most cases, these users are not merchants themselves.

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, or a retail (Card 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 Web site 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 through 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 *Standards, Compliance, and Security* video.

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 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 is a secure Web site on which merchants can manage their payment gateway account settings, including their Web site integration settings. Review the *Merchant Integration Guide* 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, it is recommended that you 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 80 of this document for a list of fields we recommend be submitted per transaction.

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

Feature	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 enable the merchant to reject transactions based on the AVS response received.	To implement AVS, merchants must require the Address and ZIP Code fields on their custom payment form. For more information about AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Table 1 Features of AIM (Continued)

Feature	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 enable the merchant to reject transactions based on the CCV response received.	To implement CCV, merchants must require the Card Code on their custom payment form. For more information about CCV, see the *Merchant Integration Guide* 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 merchant transaction confirmation 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 the "Itemized Order Information," page 43 section of this document for details.
Receipt	This feature enables merchants to opt for an automatic receipt to be sent by the payment gateway to their customers.	To configure the payment gateway receipt, merchants must require customer addresses on their custom payment form, and settings must be configured in the Receipts section of the Settings menu in the Merchant Interface or submitted for each transaction. See the "Email Receipt," page 75 section of this document for details.

Payment Processors

The merchant's payment processor determines the card types and currencies that the merchant can support.

North American Payment Processors

Authorize.Net supports the following payment processors.

Table 2 North American Payment Processors, Accepted Card Types, and Accepted Currencies

Payment Processor	Accepted Card Types	Accepted Currencies
Chase Paymentech Tampa	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD) Canadian Dollar (CAD)
Elavon	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD) Canadian Dollar (CAD)
EVO Payments	■ American Express ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD)
First Data Merchant Services (FDMS) Omaha, Nashville, and EFSNet	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD)

Table 2 North American Payment Processors, Accepted Card Types, and Accepted Currencies (Continued)

Payment Processor	Accepted Card Types	Accepted Currencies
Global Payments	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD) Canadian Dollar (CAD)
Heartland Payment Systems	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD)
TSYS Acquiring Solutions	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD)
WorldPay Atlanta	■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa	United States Dollar (USD)

European Payment Processors

Authorize.Net supports the following European payment processors.

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

Payment Processor	Accepted Card Types	Accepted Currencies
AIB Merchant Services	■ Mastercard ■ Visa	British Pounds (GBP) Euro (EUR) United States Dollar (USD)

Table 3 European Payment Processors, Accepted Card Types, and Accepted Currencies (Continued)

Payment Processor	Accepted Card Types	Accepted Currencies
Barclaycard	■ JCB ■ Mastercard ■ Visa	British Pounds (GBP) Euro (EUR)
First Data Merchant Solutions (MSIP platform)	■ Mastercard ■ Visa	British Pounds (GBP)
HSBC Merchant Services	■ Mastercard ■ Visa	British Pounds (GBP) Euro (EUR) United States Dollar (USD)
Lloyds Bank Cardnet	■ Mastercard ■ Visa	British Pounds (GBP)
Streamline	■ JCB ■ Mastercard ■ Visa	British Pounds (GBP) Euro (EUR) United States Dollar (USD)

Asia-Pacific Processors

Authorize.Net supports the following Asia-Pacific payment processors 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	■ Mastercard ■ Visa	Australian Dollar (AUD) New Zealand Dollar (NZD) United States Dollar (USD)
Westpac	■ Mastercard ■ Visa	Australian Dollar (AUD)

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. **In the event that 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* at http://developer.authorize.net/guides/echeck.pdf for more information.

Note

This feature applies only to e-commerce applications.

PayPal Express Checkout

You can use AIM to implement PayPal Express Checkout as an alternative payment. For more information see *PayPal Express Checkout Services Using AIM XML*.

Using an Encrypted Card Reader

Encrypted card readers encrypt sensitive cardholder data at the device read-head using 3DES and DUKPT encryption keys. As a result, sensitive cardholder data is transmitted without the application, the mobile device, or any other device in the data transmission chain being able to access it. This simplifies PCI compliance.

Encrypted card readers can only be used by Authorize.Net merchants with a Card Present (retail) account. Card Present accounts can still key in cardholder data on an exception basis. Exceptions are set by your acquiring bank, and may include:

- Unreadable Magstripe
- Damaged card
- Occasional telephone order

Merchants using an encrypted card reader qualify for Card Present transaction rates. For more information on transaction rates, contact your bank or your Authorize.Net sales contact.

For information about using the card reader to accept payments, see the **encryptedTrackData** section of "createTransactionRequest Elements," page 31.

Supported Card Readers

The payment platform and this API are designed to support card readers that use Visa-specified encryption keys. For more information, see "Supported Card Readers," page 94

Developer Support

Resources are available to help you successfully integrate a merchant web site or other application to the Authorize.Net Payment Gateway.

- The Developer Center provides sandbox accounts, sample code, FAQs, and troubleshooting tools.
- Developer training videos cover a variety of topics.
- The developer community provides answers to questions from other Authorize.Net developers.
- Ask us a question at our Developer Support page.
- Search our knowledge base for answers to commonly asked questions.

To submit suggestions for improving or correcting this guide, send email to documentation@authorize.net.

CHAPTER 2

Submitting Transactions

To implement AIM for a merchant's Web site or proprietary business application, you 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.

There are two options for developing the application:

- You can develop a custom application yourself using the information provided in this document, OR
- You can use Authorize.Net sample code available for free from our Developer Center at http://developer.authorize.net/downloads/samplecode/.

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 fast, 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.

Web Service Locations

Item	Location
Production	https://api.authorize.net/xml/v1/request.api
Developer Test	https://apitest.authorize.net/xml/v1/request.api
XML Schema	https://api.authorize.net/xml/v1/schema/AnetApiSchema.xsd

In order to be processed successfully, API requests and responses must conform to the AIM API XML schema.


Note

The Developer Test URL requires the use of a developer test payment gateway account. You can request a test account from our Developer Center: http://developer.authorize.net/testaccount. Developer test accounts cannot be used to test against the Production URL.

Note for .NET programmers: When a parameter is optional, and you are using serialization, the .NET language you are using automatically creates Boolean properties that indicate whether or not non-nullable parameters are specified. For example, for a parameter named validationMode that is an Enumeration type, a parameter called `validationModeSpecified` will automatically be created. By default, these properties are set to "false." If a request passes a value for an optional parameter, be sure to set these properties to "true" so that the value is not ignored.

Minimum Required Elements

The following table represents the minimum XML elements required for submitting a credit card transaction request to the payment gateway using AIM.

Table 5 Minimum Required Elements

Field Name	Value	Format	Notes
name	The merchant's unique API Login ID	Up to 20 characters	The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant the authentication required for access to the payment gateway. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for more information.

Table 5 Minimum Required Elements (Continued)

Field Name	Value	Format	Notes
transactionKey	The merchant's unique Transaction Key	16 characters	The merchant Transaction Key is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant the authentication required for access to the payment gateway. This field is not used for mobile devices. Use the mobileDeviceId field instead. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for more information.
mobileDeviceId	The unique identifier for a mobile device	60 characters	This field is required only for mobile devices accessing the gateway.
transactionType	The type of credit card transaction	authCaptureTransaction (default), authOnlyTransaction, captureOnlyTransaction, refundTransaction, priorAuthCaptureTransaction, voidTransaction	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 authCaptureTransaction.
amount	The amount of the transaction	Up to 15 digits with a decimal point (no dollar symbol) Ex. 8.95	This is the total amount and must *include* tax, shipping, and any other charges.

Table 5 Minimum Required Elements (Continued)

Field Name	Value	Format	Notes
track1 (retail applications only)	Conditional Required only if track2, cardNumber, and expirationDate are absent.	Valid Track 1 data **Note** Starting and ending sentinel characters must be discarded before submitting transactions.	Track 1 data read from credit card. This information is required only if Track 2 data and cardNumber and expirationDate are absent. It is not necessary to submit Track 1 *and* Track 2 data *and* cardNumber and expirationDate. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but cardNumber and expirationDate are submitted, the Card Present transaction rate may be downgraded.
track2 (retail applications only)	Conditional Required only if track1, cardNumber, and expirationDate are absent.	Valid Track 2 data **Note** Starting and ending sentinel characters must be discarded before submitting transactions.	Track 2 data read from credit card. This information is required only if Track 1 and cardNumber and expirationDate are absent. It is not necessary to submit Track 1 *and* Track 2 data *and* cardNumber and expirationDate. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but cardNumber and expirationDate are submitted, the Card Present transaction rate may be downgraded.
cardNumber	The customer's credit card number For Card Present (retail) applications, this is required when Track 1 or Track 2 data is absent or for manually entered transactions and refund transactions (transactionType = refundTransaction)	Between 13 and 16 digits without spaces When x_type=CREDIT, only the last four digits are required.	This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. For more information about PCI, see the *Developer Security Best Practices White Paper*: http://www.authorize.net/files/developerbestpractices.pdf

Table 5 Minimum Required Elements (Continued)

Field Name	Value	Format	Notes
expirationDate	The customer's credit card expiration date For Card Present applications, this is required when Track 1 or Track 2 data is absent or for manually entered transactions and refund transactions (transactionType = refundTransaction)	MMYY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY	This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. For more information about PCI, see the *Developer Security Best Practices White Paper*: http://www.authorize.net/files/developerbestpractices.pdf
refTransId	The payment gateway assigned transaction ID of an original transaction		Required only for `refundTransaction`, `priorAuthCaptureTransaction`, and `voidTransaction`. Do not include this field if you are providing `splitTenderId`. For more information about transaction types, see the "Credit Card Transaction Types," page 22 section of this document.
splitTenderId	The payment gateway-assigned ID that links the current authorization request to the original authorization request.	Numeric	This value applies to partial authorization transactions only, and is returned in the reply message from the original authorization request. For more information, see "Partial Authorization Transactions," page 27.
authCode	The authorization code of an original transaction *not* authorized on the payment gateway	6 characters	Required only for `captureOnlyTransaction` See the "Credit Card Transaction Types" section below.

Conditional Required Fields

The following fields are required when using GPN Canada and WorldPay Streamline Processing:

- address
- city
- country
- email
- firstName
- LastName
- state
- zip

For more information, see "createTransactionRequest Elements" on page 31.

Credit Card Transaction Types

Note

When using a European Payment Processor, additional fields are required. For more information, see "createTransactionRequest Elements" on page 31.

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

This section describes the credit card transaction types supported by the payment gateway and their unique XML requirements. It's a good idea to talk to your merchant about how their business plans to submit transactions 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 credit card transaction type is part of the `<transactionRequest>` element, which includes all of the elements listed in the "createTransactionRequest Elements" table on page 31.

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 80.

For examples of how these transaction types fit into the XML structure, see Appendix B, "Request and Response Example," on page 82.

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 value for an Authorization and Capture transaction is:

```
<transactionType>authCaptureTransaction</transactionType>
```

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 value for an Authorization Only transaction is:

```
<transactionType>authOnlyTransaction</transactionType>
```

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 in the event that 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 transaction together are considered one complete transaction. When 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 errored.
- 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 element values for a Prior Authorization and Capture transaction are:

```
<transactionType>priorAuthCaptureTransaction</transactionType>
```

In addition, the transaction ID of the original transaction needs to be specified in the `<refTransId>` element:

```
<refTransId>123456</refTransId>
```

For this transaction type, the amount field (`<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 element values for a Capture Only transaction are:

```
<transactionType>captureOnlyTransaction</transactionType>

<authCode>authorization code here </authCode>
```

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 (Refunds) if the following conditions are met:

- The transaction is submitted with the valid transaction ID (`<refTransId>`) 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 (Refund) 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 (`<cardNumber>`) 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 element values for a Credit (Refund) are:

```
<transactionType>refundTransaction</transactionType>

<refTransId> transaction ID here</refTransId>

<cardNumber>Full credit card number or last 4 digits here</cardNumber>
```

Unlinked Credit (Refund)

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 (refunds) is not a standard feature of a merchant's payment gateway account. To be enabled for expanded credit capability (ECC), the merchant must submit an application. For more information about the ECC application, see http://www.authorize.net/files/ecc.pdf.

Important

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

The unique element value for an Unlinked Credit (Refund) transaction is:

```
<transactionType>refundTransaction</transactionType>
```

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 transaction. 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 fails, 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 (`<refTransId>`) of an original, successfully authorized transaction. To void an entire order, submit the split tender ID (`<splitTenderId>`.
- The original transaction is not already settled, expired, or failed.

The unique element value for a Void is:

```
<transactionType>voidTransaction</transactionType>
```


Note

Typically, Authorization Only or Authorization and Capture are the primary transaction types submitted by an e-commerce Web site or other application. They most likely will not be used for the merchant's Web site integration, but 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

This section does not apply to retail (Card Present) transactions.

The following are required for $0 Visa verification calls:

- The transaction type must be Authorization Only. All other transaction types will be rejected.
- Bill To address and zip code are required in order to perform the AVS check. These components are found in the <billTo> element and are <address> and <zip>, respectively.

Not all processors accept a $0 amount.


Note

The payment processor EVO does not support Visa Verification transactions.

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 `<allowPartialAuth>true</allowPartialAuth>` 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 `<splitTenderId>value</splitTenderId>`. 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 `priorAuthCaptureTransaction` 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, 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 `<allowPartialAuth>true</allowPartialAuth>` 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 is complete (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 elements that apply to Partial Authorization transactions are:

- `<allowPartialAuth>true</allowPartialAuth>` (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.
- `<balanceOnCard> (output)`—this is the authorized amount remaining on the card.
- `<requestedAmount> (output)`—this is the amount requested.
- `<splitTenderId> (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.

- `<splitTenderStatus> (output)`—indicates whether or not the transaction is complete.
- `<accountType> (output)`—the card type.

Note

The payment processor EVO does not support partial authorizations.

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.

CHAPTER 3

Executing an API Call

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.


Note

If you are using an Authorize.Net developer test account, test transactions are posted to a staging environment at https://apitest.authorize.net/xml/v1/request.api. If you do not have a developer test account, you can sign up for one at http://developer.authorize.net.

Submitting Transactions Using the AIM API

The `<createTransactionRequest>` XML type has the following elements; clicking on an item displays the row that describes that element:

- merchantAuthentication
- transactionType
- amount
- payment
- authCode
- refTransId
- splitTenderId
- order
- lineItems
- tax
- billTo
- shipping
- taxExempt
- poNumber
- customer

- billTo
- shipTo
- customerIP
- retail
- cardholderAuthentication
- transactionSettings

In addition, up to 20 user fields are allowed; these are discussed in "Merchant-Defined Fields," page 46.

You can see a request example in Appendix B, "Request and Response Example," on page 82.

The following table lists the transaction data that can be submitted using the `<createTransactionRequest>` element.

createTransactionRequest Elements

Table 6 createTransactionRequest Elements

Element	Value	Format	Notes
merchantAuthentication			
name	Required. Merchant's unique API Login ID	Up to 20 characters	The merchant API Login ID is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for more information.
transactionKey	Required. Merchant's unique Transaction Key	16 characters	The merchant Transaction Key is provided in the Merchant Interface and must be stored securely. The API Login ID and Transaction Key together provide the merchant authentication required for access to the payment gateway. See the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ for more information.
mobileDeviceId	Unique identifier for a mobile device	60 characters	This field is required only for mobile devices accessing the gateway.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
refId	Optional Merchant-assigned reference ID for the request	Up to 20 characters	If included in the request, this value will be included in the response. This feature might be especially useful for multi-threaded applications. This element is not used for mobile device requests. Use mobileDeviceId instead.
employeeId	Merchant-assigned employee ID.	Numeric, 4 digits.	Required only if your payment processor is EVO.
transactionRequest This element includes all of the fields that follow, to the end of this table.			
transactionType	Optional Type of credit card transaction	authOnlyTransaction authCaptureTransaction captureOnlyTransaction refundTransaction priorAuthCapture Transaction voidTransaction	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 authCaptureTransaction.
amount	Required Amount of the transaction.	Up to 15 digits with a decimal point (no dollar symbol) For example, 8.95	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.
payment	This section includes payment information.		
trackData	trackData can contain track1 and track2		
track1	Conditional Required only if track2, cardNumber, and expirationDate are absent.	Valid Track 1 data **Note** Starting and ending sentinel characters must be discarded before submitting transactions.	Track 1 data read from credit card. It is not necessary to submit Track 1 *and* Track 2 data, *and* cardNumber and expirationDate. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but cardNumber and expirationDate are submitted, the Card Present transaction rate may be downgraded.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
track2	(Applies to Card Present only.) Conditional Required only if track1 and cardNumber, and expirationDate are absent.	Valid Track 2 data **Note** Starting and ending sentinel characters must be discarded before submitting transactions.	Track 2 data read from credit card. This information is required only if Track 1 and cardNumber and expirationDate are absent. It is not necessary to submit Track 1 *and* Track 2 data, *and* cardNumber and expirationDate. If both tracks are sent by the POS application, the gateway will use the Track 1 information. If neither Track 1 nor Track 2 data is submitted, but cardNumber and expirationDate are submitted, the Card Present transaction rate may be downgraded.
encryptedTrackData	The following elements are required only when using an encrypted card reader. For more information, see "Supported Card Readers," page 94.		
FormOfPayment			Contains the **value** element.
Value			Contains the **encoding** element.
Encoding	Hex or Base64	String	Used to specify the encoding for data in the request. If the response contains decrypted data, then it will be encoded using this setting.
EncryptionAlgorithm	TDES	String	Specifies the algorithm used to encrypt the payment data. Currently, only TDES is supported.
Scheme			Contains the **DUKPT** element.
DUKPT			Contains the **Operation** element.
Operation	DECRYPT	String	Used to specify the cryptographic operation to perform on the request data. Currently, only **Decrypt** is supported and must be specified.
Mode			Contains the **Data** element.
Data	1	Boolean. 1 means true and is currently the only value supported.	Used to specify that DUKPT decrypt
DeviceInfo			Contains the **Description** element.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
Description	512 encoded characters	String	Metadata string used to control how the request is ultimately processed. Contains key value pairs with the following format: KEY_NAME=KEY_VALUE Where • KEY_NAME is an alphanumeric identifier and may contain a period (`.`), hyphen (`-`), and underscore (`_`). • `=` separates a key name and its value. • KEY_VALUE is the value of the preceding key. It cannot contain the following special characters: `\ , ; = < > % ^` The `\` character can be used to escape special characters. The ENCODING that is used is based on the Encoding set in the request. Metadata is used to specify how the request should be processed. For an example in plain ASCII: FID=IDTECH.UniMag.Android.Sdkv1 Note that this string can be used with any supported platform, despite the term “android.” Hex Encoded: 4649443d4944544543482e556e694d61672e416e64726f69642e53646b7631 Base64 Encoded: 434f4d4d4f4e2e456e63727970746564547261636b732e53646b7631 For more information, see "Supported Card Readers," page 94.
EncryptedData			Contains **Value** element.
Value	Encrypted card data.	String	Encoded card reader data that contains encrypted tracks and KSN.
creditCard	The following elements belong to the <creditCard> element; include them only for credit card transactions.		

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
cardNumber	Required Optional for Card Present The customer's credit card number	Between 13 and 16 digits without spaces Only the last four digits are required for credit card transactions.	This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. For more information about PCI, see the *Developer Security Best Practices White Paper* at http://www.authorize.net/files/developerbestpractices.pdf.
expirationDate	Required Optional for Card Present The customer's credit card expiration date	MMYY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY	This is sensitive cardholder information and must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. For more information about PCI, see the *Developer Security Best Practices White Paper* at http://www.authorize.net/files/developerbestpractices.pdf.
cardCode	Optional The customer's card code	Numeric	The three- or four-digit number on the back of a credit card (on the front for American Express). This field is required if the merchant would like to use the Card Code Verification (CCV) security feature. For more information, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ . Cardholder information must be stored securely and in accordance with the Payment Card Industry (PCI) Data Security Standard. See the *Developer Security Best Practices White Paper* at http://www.authorize.net/files/developerbestpractices.pdf for more information.
bankAccount	The following elements belong to the <bankAccount> element; include them only for bank account transactions.		
routingNumber	Routing number for bank	XXXX0000	
accountNumber	Account number, masked	XXXX1111	

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
nameOnAccount			
bankName			
echeckType		PPD, WEB, CCD, TEL, ARC, BOC	
checkNumber	The check number as printed on the physical check.	Numeric	Only applicable to ARC and BOC eCheck types.
authCode	Optional Authorization code	string	
refTransId	Optional Transaction ID of original partial authorization transaction.	string	Used only for second and subsequent partial authorization transactions.
splitTenderId	Optional		
order	Contains information about order.		
invoiceNumber	Merchant-defined invoice number associated with order.		
Description	Description of item purchased.		
profile	The following fields enable you to charge a transaction using existing payment or shipping profiles. For more information on profiles, see the Customer Profiles Getting Started Guide.		
createProfile	true, false		Optional If set to true, a CIM profile will be generated from the customer and payment data.
customerProfileId	The CIM customer ID.		Required if you are using a CIM profile as the source for payment or shipping information.
paymentProfile	Contains payment profile information.		
paymentProfileId	The CIM payment profile ID.		Designates the payment profile to use for payment and billing information. Required if the paymentProfile element exists.
cardCode			Optional. Because card codes are not stored, they are not a part of the paymentProfileId. A merchant can choose to collect it at checkout for additional security.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
shippingProfileId	The CIM shipping profile ID.		Optional. This field is mutually exclusive with the ShipTo section. Use one or the other.
lineItems	Contains one or more <lineItem> elements (the maximum is 30 line items).		
lineItem	Describes one line item of order		
itemId	Item identification	Up to 31 characters	ID assigned to item.
name	Name of item	Up to 31 characters	Short description of item.
description	Description of item	Up to 255 characters	Detailed description of item.
quantity	Quantity purchased	Up to two decimal places Must be a positive number	Quantity of item.
unitPrice	Price of one item	Up to two decimal places Must be positive number	Cost of item per unit, *excluding* tax, freight, and duty.
tax	Contains information about any taxes applied.		
amount	Amount of tax	Format can include up to two decimal points. For example, 1.27.	Total amount of transaction must include this amount.
name	Name of tax		
description	Description of tax		
duty	Contains information about any duty applied.		
amount	Amount of duty		
name	Name of duty		
description	Description of duty		
shipping	Items in this section describe shipping charges applied.		
amount	Amount of shipping charges		
name	Name of shipping charges		
description	Description of shipping charges		
taxExempt	Optional. Indicates whether or not order is exempt from tax	true, false	
poNumber	Merchant-assigned purchase order number.	Up to 25 characters (no symbols)	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.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
customer	The following fields contain customer information.		
type		Individual, business	
id	Merchant assigned customer ID	Up to 20 characters (no symbols)	Unique identifier to represent the customer associated with the transaction. Customer ID must be created dynamically on the merchant server or provided for each transaction. The payment gateway does not perform this function.
email	Customer's valid email address	Up to 255 characters For example, janedoe@customer.com	Required only when using a European Payment Processor. Processing Platform. Email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid. For more information about Email Receipts, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ .
billTo	This section contains billing address information.		
firstName	First name associated with customer's billing address	Up to 50 characters (no symbols)	Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
lastName	Last name associated with customer's billing address	Up to 50 characters (no symbols)	Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
company	Company associated with customer's billing address	Up to 50 characters (no symbols)	

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
address	Customer's billing address	Up to 60 characters (no symbols)	Required when you use GPN Canada or WorldPay Streamline Processing Platform. Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42. Required if merchant would like to use the Address Verification Service security feature. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/
city	City of customer's billing address	Up to 40 characters (no symbols)	Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
state	State of customer's billing address	Up to 40 characters (no symbols) or a valid two-character state code	Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
zip	ZIP code of customer's billing address	Up to 20 characters (no symbols)	Required when using GPN Canada or WorldPay Streamline Processing Platform. Required when you use a European Payment Processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42. Required if merchant would like to use the Address Verification Service security feature. For more information on AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/ .

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
country	Country of customer's billing address	Up to 60 characters (no symbols)	Required only when using a European Payment Processor.
phoneNumber	Phone number associated with customer's billing address	Up to 25 digits (no letters) For example, (123)123-1234	
faxNumber	Fax number associated with customer's billing address	Up to 25 digits (no letters) For example, (123)123-1234	
shipTo	This section contains shipping information.		
firstName	Optional First name associated with customer's shipping address	Up to 50 characters (no symbols)	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
lastName	Optional Last name associated with customer's shipping address	Up to 50 characters (no symbols)	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
company	Optional Company associated with customer's shipping address	Up to 50 characters (no symbols)	
address	Optional Customer's shipping address	Up to 60 characters (no symbols)	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
city	Optional City of customer's shipping address	Up to 40 characters (no symbols)	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
state	Optional State of customer's shipping address	Up to 40 characters (no symbols) or a valid two-character state code	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.
zip	Optional ZIP code of customer's shipping address	Up to 20 characters (no symbols)	If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Information," page 42.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
country	Optional Country of customer's shipping address	Up to 60 characters (no symbols)	
customerIP	IP address of customer initiating the transaction. If this value is not passed, it defaults to 255.255.255.255.	Up to 15 characters (no letters) For example, 255.255.255.255	Required only when the merchant is using customer IP based AFDS filters
cardholderAuthentication	Merchants using a third party cardholder authentication solution can submit the following authentication values with Visa and MasterCard transactions. **Note** invalid combinations of the following two fields will generate an error. For more information, read the section "Cardholder Authentication," page 44.		
Authentication Indicator	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.	Special characters included in this value must be URL encoded.	Required only for transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored. This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.
Cardholder Authentication 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.	Special characters included in this value must be URL encoded.	Required only for authOnly and authCapture transactions processed through cardholder authentication programs. When submitted with other transaction types, this value is ignored. This field is currently supported through Chase Paymentech, FDMS Nashville, Global Payments and TSYS.

Table 6 createTransactionRequest Elements (Continued)

Element	Value	Format	Notes
retail	The retail element contains two elements: marketType and deviceType.		
marketType	0 for ecommerce 1 for moto 2 for retail		The default for Card Present accounts is 2 and only 2 can be used. For blended accounts, the default is 0 and marketType can be overridden.
deviceType	7 for mobile POS		
transactionSettings	This section contains one or more <setting> elements		
setting	The element consists settingName and settingValue fields.		
settingName	Option being set	allowPartialAuth, duplicatWindow, emailCustomer, recurringBilling, testRequest	
settingValue	true or false		
userFields	User-defined fields are allowed. For information about how to use them, see "Merchant-Defined Fields," page 46.		
name	Name of user-defined field		
value	Value of user-defined field		

EVO Billing and Shipping Information

If your payment processor is EVO and you submit one of the following **billTo** fields, you must submit all of them.

- firstName
- lastName
- address
- city
- state
- zip

If your payment processor is EVO and you submit one of the following **shipTo** fields, you must submit all of them.

- firstName
- lastName
- address
- city

- state
- zip

Itemized Order Information

Based on their respective business requirements, merchants can choose to submit itemized order information with a transaction. Itemized order information is not submitted to the processor and is 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 merchant can submit up to 30 distinct line items containing itemized order information per transaction.

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.

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

Cardholder Authentication

The payment gateway accepts authentication fields for the following programs:

- Verified by Visa
- MasterCard® SecureCode™

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

Note

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

Invalid combinations of the `authenticationIndicator` and `cardholderAuthenticationValue` fields cause the transaction to generate an error.

Valid value combinations for these fields are as follows:

Visa

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

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

For example, when the MasterCard value for `authenticationIndicator` is “1,” the

value for `cardholderAuthenticationValue` must be null. In this scenario, if a value is submitted for `cardholderAuthenticationValue`, 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.

Customer Transaction Receipts

With the payment gateway you can send customer transaction receipts for an approved transaction. The method works for transactions that were approved, were not voided, and are not expired.

The email is not sent if a merchant does not have a “receipt reply to” address set up in their account.

sendCustomerTransactionReceiptRequest

Element	Value	Format
merchantAuthentication	Contains the merchant’s payment gateway account authentication information	
name	The merchant’s unique API Login ID	Up to 20 characters
transactionKey	The merchant’s unique Transaction Key	16 characters
refId	Merchant-assigned reference ID for the request	Up to 20 characters
transId	The payment gateway assigned identification number for the transaction	When testRequest is set to a positive response, or when Test Mode is enabled on the payment gateway, this value will be “0.”
customerEmail	The customer’s valid email address	Up to 255 characters For example, janedoe@customer.com
emailSettings		
setting		
settingName	Name of the desired setting	string
settingValue	Value of the setting	string

Example

```
<?xml version="1.0" encoding="utf-8"?>
<sendCustomerTransactionReceiptRequest
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
      <merchantAuthentication>
          <name>MyLoginName</name>
          <transactionKey>MyTransKey</transactionKey>
```

```
      </merchantAuthentication>
      <refId>123456</refId>
      <transId>1234567890</transId>
      <customerEmail>somebody@somewhere.com</customerEmail>
      <emailSettings>
         <setting>
            <settingName>headerEmailReceipt</settingName>
            <settingValue><![CDATA[<html><head></head><body>some
HEADER stuff</body></html>]]></settingValue>
         </setting>
         <setting>
            <settingName>footerEmailReceipt</settingName>
            <settingValue><![CDATA[<html><head></head><body>some
FOOTER stuff</body></html>]]></settingValue>
         </setting>
      </emailSettings>
</sendCustomerTransactionReceiptRequest>
```

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*.

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). In the event that Authorize.Net, a CyberSource solution, discovers that a 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.

Mobile Device Transactions

These calls are used to register a mobile or POS device and submit transactions to the Authorize.Net system. Unlike standard IP-based transactions, the mobile API calls enable you to authenticate and receive a session token for transactions and reporting calls, thus eliminating the need to store credentials on the device. Additionally, a device registered with Authorize.Net can be disabled if it is lost or stolen.

When a mobile device is registered and approved, it can be used to transmit transaction information using the Merchant Web Services API. Use the functions described in this section to log in and out, and to start a session.

The following functions are available:

- "mobileDeviceRegistrationRequest"
- "mobileDeviceLoginRequest"
- "logoutRequest"

The merchantAuthentication block is different when the request comes from a mobile device. See the following sections for examples.

mobileDeviceRegistrationRequest

This function initially registers a mobile device. If the device is successfully inserted for approval, the user receives an OK message, I00005.

- If the device is already in the database but pending, the client receives error E00055. If the merchant already has 100 pending devices, the client receives error E00058.
- If the device is already approved and ready for use, the client receives informational message I00006.

Table 7 Fields in the mobileDeviceRegistrationRequest call

Element	Value	Format
merchantAuthentication	Contains the merchant's payment gateway account authentication information	
name	The merchant's Login ID.	Up to 20 characters
password	Password for the merchant's Login ID.	
mobileDevice	Contains information about mobile devices	
mobileDeviceId	String	Maximum length 60
description	String	Maximum length 60
phoneNumber	String	Maximum length 20

Example

```
<?xml version="1.0" encoding="utf-16"?>
<mobileDeviceRegistrationRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>Merchant's MINT Username</name>
    <password>Merchant_MINT_Password</password>
  </merchantAuthentication>
  <mobileDevice>
    <mobileDeviceId>AHJFDJNEOIIOU893457DJHG</mobileDeviceId>
    <description>Floor Employee - John</description>
    <phoneNumber>(206) 555-1234</phoneNumber>
  </mobileDevice>
</mobileDeviceRegistrationRequest>
```

mobileDeviceLoginRequest

This is the first request from a mobile device to start a session. The request requires a **merchantAuthenticationType** object which would include a name (AnetUserName), password, and mobile device ID (mobileDeviceId). If the mobile device ID is not registered, the client receives error E00056. If the mobile device ID has been submitted but is pending, the client receives error E00055. If the device is disabled, the client receives error E00056. If the user name and password combination is wrong, the client receives standard error E00007.

The **mobileDeviceId** is expected to be in the **mobileDeviceType** object as opposed to the **merchantAuthenticationType** object.

If the username and password combination is correct, and the mobile device is approved, the client receives the message OK, as well as a collection of permissions associated with the user, a session token, and the merchant contact information.

- The session token will time out after 60 minutes of inactivity and will remain valid for a maximum of 8 hours.

Element	Value	Format
merchantAuthentication	Contains the merchant's payment gateway account authentication information	
name	The merchant's unique API Login ID	Up to 20 characters
password	Password for the account	
mobileDeviceId	String	Maximum length 60

Example

```
<?xml version="1.0" encoding="utf-16"?>
<mobileDeviceLoginRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>Merchant Name</name>
    <password>merchant_password</password>
    <mobileDeviceId>mpldf58693</mobileDeviceId>
  </merchantAuthentication>
</mobileDeviceLoginRequest>
```

logoutRequest

This object transmits a request to log out of the system.

Example

```
<?xml version="1.0" encoding="utf-16"?>
<logoutRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>Merchant Name</name>
    <sessionToken>HS1bSblcK1Ra3DxFNMP6Hx6nRAryQAFPPVJxMRMwmdEA
</sessionToken>
    <mobileDeviceId>mpldf58693</mobileDeviceId>
  </merchantAuthentication>
</logoutRequest>
```

createTransactionRequest

A request from a mobile device to create a transaction is identical to the general createTransactionRequest call, with the exception of the merchant authentication block, which includes:

- Name
- Session token
- Mobile Device ID

One other optional exception is when an encrypted card read is used. For a code example that shows encrypted card reader fields, see "Encrypted Mobile Card Reader Track Data," page 85.

For more information on encrypted card readers, see "</createTransactionRequest>," page 51.

For a complete list of fields in a transaction request, see "createTransactionRequest Elements," page 31.

Example createTransactionRequest

```
<?xml version="1.0" encoding="utf-16"?>
<createTransactionRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>Merchant Name</name>
<sessionToken>gAc9F$cY0VRqpzLLlfhWdhQYR9WWLxhUvnXWSE9ffqkA</sessionToken>
    <mobileDeviceId>mpldf58693</mobileDeviceId>
  </merchantAuthentication>
  <transactionRequest>
    <transactionType>authCaptureTransaction</transactionType>
    <amount>10.00</amount>
    <payment>
      <creditCard>
        <cardNumber>5424000000000015</cardNumber>
        <expirationDate>0511</expirationDate>
        <cardCode>123</cardCode>
      </creditCard>
    </payment>
    <order>
      <invoiceNumber>INV001</invoiceNumber>
      <description>Really nice things!</description>
    </order>
    <billTo>
      <firstName>John</firstName>
      <lastName>Lennon</lastName>
      <company>The Beatles</company>
      <address>1969 Abby Road</address>
      <city>Liverpool</city>
      <state>EN</state>
      <zip>UK5567</zip>
      <country>United Kingdom</country>
     <phoneNumber>555-648-9756</phoneNumber>
      <faxNumber>555-648-9757</faxNumber>
    </billTo>
    <shipTo>
      <firstName>Ringo</firstName>
      <lastName>Starr</lastName>
      <company>The Beatles</company>
      <address>1969 Penny Lane</address>
      <city>Liverpool</city>
```

```
        <state>EN</state>
        <zip>UK5567</zip>
        <country>United Kingdom</country>
      </shipTo>
      <customerIP>fe80::f4b6:2a88:70fa:f09f%13</customerIP>
      <transactionSettings>
        <setting>
          <settingName>allowPartialAuth</settingName>
          <settingValue>False</settingValue>
        </setting>
        <setting>
          <settingName>testRequest</settingName>
          <settingValue>FALSE</settingValue>
        </setting>
      </transactionSettings>
      <userFields>
        <userField>
          <name>x_type</name>
          <value>AUTH_CAPTURE</value>
        </userField>
      </userFields>
    </transactionRequest>
</createTransactionRequest>
```

CHAPTER 4

Transaction Response

The transaction response from the payment gateway is returned in the `<createTransactionResponse>` element. This element indicates the status of a transaction.

Fields in the Payment Gateway Response

The following table lists the fields returned in the response from the payment gateway. You can see a response example in Appendix B, "Request and Response Example," on page 82.

There are two <message> elements: one is inside the <messages> element at the beginning of the response and describes the status of your request; the other is within the <transactionResponse> element and describes the status of the original transaction. Both are described in Appendix C, "Information and Error Messages," on page 87.

Table 8 createTransactionResponse Fields

Element	Value	Format	Notes
refId	Merchant-assigned reference ID for the request *Optional*	Up to 20 characters	If included in the request, this value is included in the response. This feature might be especially useful for multi-threaded applications.
messages	This section contains information about the results of the request.		
resultCode		Ok Error	Contains additional information about the status of the request.
message	Contains specific message information		
code	Code number for message	I00001 E000001	For an explanation of error codes, see Appendix C, "Information and Error Messages," on page 87.

Table 8 createTransactionResponse Fields (Continued)

Element	Value	Format	Notes
text	Text for error message		
sessionToken			Returned for mobile device transactions, instead of a transactionKey
transactionResponse			
responseCode	Overall status of transaction	1 = Approved 2 = Declined 3 = Error 4 = Held for Review	
authCode	Authorization or approval code	6 characters	
avsResultCode	Address Verification Service (AVS) response code	A = Address (Street) matches, ZIP does not B = Address information not provided for AVS check E = AVS error G = Non-U.S. Card Issuing Bank N = No Match on Address (Street) or ZIP P = AVS not applicable for this transaction R = Retry—System unavailable or timed out S = Service not supported by issuer U = Address information is unavailable W = Nine digit ZIP matches, Address (Street) does not X = Address (Street) and nine digit ZIP match Y = Address (Street) and five digit ZIP match Z = Five digit ZIP matches, Address (Street) does not	Indicates the result of the AVS filter. For more information about AVS, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

Table 8 createTransactionResponse Fields (Continued)

Element	Value	Format	Notes
cvvResultCode	Card code verification (CCV) response code	M = Match N = No Match P = Not Processed S = Should have been present U = Issuer unable to process request	Indicates result of the CCV filter. For more information about CCV, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.
cavvResultCode	Cardholder authentication verification response code		Blank or not present = CAVV not validated 0 = CAVV not validated because erroneous data was submitted 1 = CAVV failed validation 2 = CAVV passed validation 3 = CAVV validation could not be performed; issuer attempt incomplete 4 = CAVV validation could not be performed; issuer system error 5 = Reserved for future use 6 = Reserved for future use 7 = CAVV attempt—failed validation—issuer available (U.S.-issued card/non-U.S acquirer) 8 = CAVV attempt—passed validation—issuer available (U.S.-issued card/non-U.S. acquirer) 9 = CAVV attempt—failed validation—issuer unavailable (U.S.-issued card/non-U.S. acquirer) A = CAVV attempt—passed validation—issuer unavailable (U.S.-issued card/non-U.S. acquirer) B = CAVV passed validation, information only, no liability shift
transId	Payment gateway assigned identification number for transaction	When testRequest is set to a positive response, or when Test Mode is enabled on the payment gateway, this value will be 0.	This value must be used for any follow-on transactions such as a credit, prior auth capture, or void.

Table 8 createTransactionResponse Fields (Continued)

Element	Value	Format	Notes
reftransId			Optional. The transaction ID of a related, previously settled transaction.
transHash	Payment gateway-generated MD5 hash value that can be used to authenticate transaction response.	Alphanumeric	Because transaction responses are returned using an SSL connection, this feature is not necessary for AIM.
testRequest		true, false 1,0	Indicates whether or not to treat this request as a test transaction.
accountNumber			
accountType			
messages	This element contains one or more <message> elements		
message	These messages contain detailed information about the status of a particular transaction.		
code		Response code that represents status	For a complete list of response codes and descriptions, see the "Response Code Details," page 61 section.
description		Text description of status	
errors	This element contains one or more <error> elements		
error	This element contains detailed information about any errors returned.		
errorCode		Error code returned	
errorText		Text description of error	
splitTenderPayments	If the transaction was a partial authorization transaction, the split tender payment detail information is listed in this section.		
splitTenderPayment	Contains information about one split tender transaction		
transId	The payment gateway assigned identification number for the transaction	When testRequest is set to a positive response, or when Test Mode is enabled on the payment gateway, this value will be “0.”	
responseCode			
responseToCustomer			
authCode			
accountNumber	Last 4 digits of card provided	Alphanumeric (XXXX6835)	This field is returned with all transactions.
accountType	Visa, MasterCard, American Express, Discover, Diners Club, JCB	Text	

Table 8 createTransactionResponse Fields (Continued)

Element	Value	Format	Notes
requestedAmount	Amount requested in original authorization	Numeric	Present if the current transaction is for a prepaid card or if a splitTenderId was sent.
approvedAmount	Amount approved		Present if the current transaction is for a prepaid card or if a splitTenderId was sent in.
balanceOnCard	Balance on debit card or prepaid card	Numeric	Can be a positive or negative number. Has a value only if the current transaction is for a prepaid card
userFields	This element contains user fields if any are defined.		
name		Name of user-defined field	These values are only echoed back in the response, and are also added to the merchant receipts. No other action is taken with user-defined fields.
value		Value of user-defined field	
profileResponse			Contains the result of an attempt to create a CIM profile using the **createTransactionRequest** method.
message			Contains detailed information about the status of a particular transaction.
code	Response Code that represents status.		For a complete list of response codes and descriptions, see "Response Code Details," page 61.
text	Text description of status.		
customerProfileId	Payment gateway-assigned ID associated with the customer profile.	Numeric	
customerPayment ProfileIdList	Contains the Customer Payment Profile ID element.		
numericString	Payment gateway-assigned ID associated with the customer shipping profile.	Numeric	This is only included if the original transaction included a shipping address.

Response for Duplicate Transactions

The AIM API enables you to specify the window of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction (based on credit card number, invoice number, amount, billing address information, transaction type, etc.) using the duplicateWindow field. 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.

sendCustomerTransactionReceiptResponse

This is a response to the **sendCustomerTransactionReceiptRequest** function.

If a request is made for a receipt related to a declined, voided, or expired transaction, the client receives the error E00060 in the `<code>` field, with appropriate message text:

Transaction state	Error message text
Declined	The transaction associated with TransactionId "[TransID]" was not approved.
Voided	The transaction associated with TransactionId "[TransID]" has been voided.
Expired	The transaction associated with TransactionId "[TransID]" has expired.

The response to the sendCustomerTransactionReceiptRequest contains the <refId> and <messages> elements, as shown in the following example:

Example

```
<?xml version="1.0" encoding="utf-8"?>
<sendCustomerTransactionReceiptResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <refId>123456</refId>
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00001</code>
      <text>Successful.</text>
    </message>
  </messages>
</sendCustomerTransactionReceiptResponse>
```

mobileDeviceRegistrationResponse

The response to the mobileDeviceRegistrationRequest function contains simply the <messages> element, as shown in the following example:

```
<?xml version="1.0" encoding="utf-16"?>
<mobileDeviceRegistrationResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00005</code>
      <text>The mobile device has been submitted for approval by the
account administrator.</text>
    </message>
  </messages>
</mobileDeviceRegistrationResponse>
```

mobileDeviceLoginResponse

The following table lists fields included in the mobileDeviceLoginResponse element.

Table 9 Mobile Device Login Response Elements

Element	Value	Format	Notes
messages	This section contains information about the results of the request.		
resultCode		Ok Error	Contains additional information about the status of the request.
message	Contains specific message information		
code	Code number for message	I00001 E000001	For an explanation of error codes, see Appendix C, "Information and Error Messages," on page 87.
text	Text for error message		
sessionToken			Returned for mobile device transactions, instead of a transactionKey
merchantContact	This section contains contact information for the merchant who registered the mobile device		
merchantName	Name of merchant		
merchantAddress	Address of merchant		
merchantCity	Merchant's city		
merchantState	Merchant's state		
merchantZip	Merchant's zip code		
merchantPhone	Merchant's phone number		
userPermissions	This section contains one or more <permission> elements.		
permission	The permissions returned apply only to the user who logged in.		
permissionName		Possible values: Submit_Charges (This user can submit charges) Submit_Refund (This user can submit refunds) Submit_Update (This user can submit updates) API_Merchant_BasicReporting (This user has basic reporting permissions) Mobile_Admin (This user has administrative permissions for the mobile device)	
merchantAccount	The following elements describe the type of merchant account.		
marketType	0 for ecommerce 1 for moto (Mail Order / Telephone ORder) 2 for retail	The default value is 2.	

Table 9 Mobile Device Login Response Elements (Continued)

Element	Value	Format	Notes
deviceType	7 for mobile POS		

Example

```
<?xml version="1.0" encoding="utf-16"?>
<mobileDeviceLoginResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00001</code>
      <text>Successful.</text>
    </message>
  </messages>
  <sessionToken>tnr6i3pLEn1wtfI055bf0OB0$bqXq6UldYwWHZrBn$oA
</sessionToken>
  <merchantContact>
    <merchantName>Business Name</merchantName>
    <merchantAddress>12345 132nd Ave NE </merchantAddress>
    <merchantCity>Seattle</merchantCity>
    <merchantState>CA</merchantState>
    <merchantZip>98006</merchantZip>
    <merchantPhone>(206) 111-2222</merchantPhone>
  </merchantContact>
  <userPermissions>
    <permission>
      <permissionName>Submit_Charge</permissionName>
    </permission>
    <permission>
      <permissionName>Submit_Refund</permissionName>
    </permission>
    <permission>
      <permissionName>Submit_Update</permissionName>
    </permission>
    <permission>
      <permissionName>API_Merchant_BasicReporting</permissionName>
    </permission>
    <permission>
      <permissionName>Mobile_Admin</permissionName>
    </permission>
  </userPermissions>
```

```
  <merchantAccount>
    <marketType>2</marketType>
    <deviceType>7</deviceType>
</mobileDeviceLoginResponse>
```

mobileDeviceLogoutResponse

The `mobileDeviceLogoutResponse` element returns only the <messages> field, as shown in the following example:

Example

```
<?xml version="1.0" encoding="utf-16"?>
<logoutResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00001</code>
      <text>Successful.</text>
    </message>
  </messages>
</logoutResponse>
```

For a list of message codes, see Appendix C, "Information and Error Messages," on page 87.

Response Code Details

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

Note

Not all of the response codes apply to this API.

- **Response Code** indicates the overall status of the transaction with possible values of approved, declined, failed, or held for review.

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

Response Codes

Table 10 Response Codes

Response Code	Description
1	This transaction has been approved.
2	This transaction has been declined.
3	There has been an error processing this transaction.
4	This transaction is being held for review.

Response Reason Codes and Response Reason Text

Table 11 Response Reason Codes and Response Reason Texts

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 `routingNumber` field did not pass validation or was not for a valid financial institution.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	10	The account number is invalid.	The value submitted in the accountNumber 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 authCode to be present was submitted without a value.
3	13	The merchant API Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applicable only to SIM and WebLink APIs.
3	15	The transaction ID is invalid.	The transaction ID value is non-numeric or was not present for a transaction that requires it (i.e., voidTransaction, priorAuthCaptureTransaction, and refundTransaction.
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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	29	The Paymentech identification numbers are incorrect. Call Merchant Service Provider.	
2	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
2	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	32	This reason code is reserved or not applicable to this API.	
3	33	*FIELD* cannot be left blank.	The word *FIELD* will be replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in. 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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	This occurs if the merchant tries to capture funds greater than the amount of the original authorization-only transaction.
3	48	This processor does not accept partial reversals.	The merchant attempted to settle for less than the originally authorized amount.
3	49	A transaction amount greater than $[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 paymentMethod = ECHECK, then transactionType cannot be set to captureOnlyTransaction.
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.	

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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 for `version` was invalid.
3	69	The transaction type is invalid.	The value submitted in `transactionType` was invalid.
3	70	The transaction method is invalid.	The value submitted in paymentMethod was invalid.
3	71	The bank account type is invalid.	The value submitted in bankAccountType was invalid.
3	72	The authorization code is invalid.	The value submitted in authCode 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 dateOfBirth was invalid.
3	74	The duty amount is invalid.	The value submitted in the `duty` element failed format validation.
3	75	The freight amount is invalid.	The value submitted in freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in the `tax` element failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in `taxId` failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in `cardCode` failed format validation.
3	79	The driver's license number is invalid.	The value submitted in the `<number>` field of the `<driversLicense>` element failed format validation.
3	80	The driver's license state is invalid.	The value submitted in the `<state>` field of the `<driversLicense>` element 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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.	
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 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 `transHash` field.
3	100	The eCheck.Net type is invalid.	Applicable only to eCheck.Net. The value specified in the `echeckType` 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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	102	This request cannot be accepted.	A password or Transaction Key was submitted with this request. This is a high security risk.
3	103	This transaction cannot be accepted.	A valid fingerprint, Transaction Key, or password is required for this transaction.
3	104	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for `country` failed validation.
3	105	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for city and country failed validation.
3	106	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for `company` failed validation.
3	107	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applicable only to eCheck.Net. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applicable only to eCheck.Net. The value submitted for bank account name does not contain valid characters.
3	116	The authentication indicator is invalid.	This error is applicable 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 `authenticationIndicator` field is invalid.
3	117	The cardholder authentication value is invalid.	This error is applicable 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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	118	The combination of authentication indicator and cardholder authentication value is invalid.	This error is applicable 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 44 section of this document.
3	119	Transactions having cardholder authentication values cannot be marked as recurring.	This error is applicable only to Verified by Visa and MasterCard SecureCode transactions. Transactions submitted with a value in `authenticationIndicator` *and* recurringBilling=YES will be rejected.
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 failed transaction. (The original transaction experienced a database error.)
3	122	An error occurred during processing. Please try again.	The system-generated void for the original failed transaction. (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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	141	This transaction has been declined.	The system-generated void for the original FraudScreen-rejected transaction failed.
2	145	This transaction has been declined.	The system-generated void for the original card code-rejected and AVS-rejected transaction failed.
3	152	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	The system-generated void for the original transaction failed. The response for the original transaction could not be communicated to the client.
2	165	This transaction has been declined.	The system-generated void for the original card code-rejected transaction failed.
3	170	An error occurred during processing. Please contact the merchant.	Concord EFS—Provisioning at the processor has not been completed.
2	171	An error occurred during processing. Please contact the merchant.	Concord EFS—This request is invalid.
2	172	An error occurred during processing. Please contact the merchant.	Concord EFS—The store ID is invalid.
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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.
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

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	213	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The merchant assessment code is incorrect.
2	214	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This function is currently unavailable.
2	215	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The encrypted PIN field format is invalid.
2	216	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ATM term ID is invalid.
2	217	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced a general message format problem.
2	218	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The PIN block format or PIN availability value is invalid.
2	219	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The ETC void is unmatched.
2	220	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The primary CPU is not available.
2	221	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The SE number is invalid.
2	222	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. Duplicate auth request (from INAS).
2	223	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. This transaction experienced an unspecified error.
2	224	This transaction has been 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 `recurringBilling` and `echeckType` is not allowed.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	244	This eCheck.Net type is not allowed for this Bank Account Type.	The combination of values submitted for `bankAccountType` and `echeckType` is not allowed.
3	245	This eCheck.Net type is not allowed when using the payment gateway hosted payment form.	The value submitted for `echeckType` 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 `transactionType` and `echeckType` is not allowed.
3	248	The check number is invalid.	Invalid check number. Check number is limited to 15 alphanumeric characters.
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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.	
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	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.

Table 11 Response Reason Codes and Response Reason Texts (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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 errors is available in our Developer Center at http://developer.authorize.net/tools/responsereasoncode.

Email Receipt

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

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

To send payment-gateway-generated customer receipts, follow these steps:

- In the **transactionSettings** element, set the **settingName** to `Customer` and set the **settingValue** to `true`.
- Enter the customer's valid email address (up to 255 characters) in the element of the **customer** field.

- To insert a header, enter **headerReceipt** in another **settingName** element. In the **settingValue** field, enter the header as it should appear in the email.
- To insert a footer, enter **footerReceipt** in another **settingName** element. In the **settingValue** field, enter the header as it should appear in the email.
- To send a confirmation to the merchant, add another **settingName** element with the value `merchant`, and set the value to the 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.

Warning

If the email is included, it can subject the merchant to spam on their business email address, because it announces where the receipt gets returned, and gives a hint where relay response information could be sent.

In addition, the merchant can receive a transaction confirmation 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 s in the Merchant Interface. For more information, see the *Merchant Integration Guide* at http://www.authorize.net/support/merchant/.

CHAPTER 5

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:

- First, use an Authorize.Net developer test account. In this environment, test transactions are posted to https://apitest.authorize.net/xml/v1/request.api. Even though 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/.

- As soon as you successfully test the integration in the developer test environment, you can enter the merchant's Authorize.Net Payment Gateway API Login ID and Transaction Key into the integration for testing in 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:

  - By including in the **transactionSettings** element a **settingName** field with the value `testRequest`, and a value of `TRUE` in the **settingValue** field.

  - 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 that you cannot test follow-on transactions such as credits and voids while in Test Mode. To test follow-on transactions, you can either submit a

settingName of `testRequest` as indicated above, or process a test transaction with any valid credit card number in live mode, as explained below.


Note

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

- If testing in the live environment is successful, you are ready to submit live transactions and verify that they are being submitted successfully. Either remove the **testRequest** element from the settings, or set it to FALSE; or, if you are using Test Mode, turn it off in the Merchant Interface. To receive a response of TRUE, 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 and zip code fields are required.

Testing to Generate Specific Transaction Results

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

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

To test the AVS or CCV responses in the live environment, you need to 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 52 of this guide.

APPENDIX A

Fields by Transaction Type

This appendix provides a complete listing of all API fields that should be submitted.. 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 in the event that integration settings in the Merchant Interface are inadvertently changed.

Minimum Required Fields

The following table provides a quick reference of all API fields that are required..

	Authorization and Capture	**Authorization Only**	**Prior Authorization and Capture**	**Capture Only**	**Credit**	**Void**
Merchant Authentication element	Name transactionKey	Name transactionKey	Name transactionKey	Name transactionKey	Name transactionKey	Name transactionKey
Transaction Type element	authCapture Transaction	authOnly Transaction	priorAuth Capture Transaction transId or refTransId or splitTenderId	captureOnlyTransaction authCode	refundTransaction refTransId	voidTransaction refTransId or splitTenderId
Payment Information	amount cardNumber expirationDate	amount cardNumber expirationDate	amount (required only when less than the original authorization amount)	amount cardNumber expirationDate	amount cardNumber expirationDate*	N/A

* For merchants with expanded credit capabilities (ECC), a Transaction ID should NOT be submitted for Unlinked Credits. For more information, see the "Credit Card Transaction Types," page 22 section of this document.
** The expiration date is required only 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 12 Required Fields for Additional AIM Features

	Authorization and Capture	Authorization Only	Prior Authorization and Capture	Capture Only	Credit	Void
Itemized Order Information	Tax Duty shipping	Tax Duty shipping	Tax Duty shipping	Tax Duty shipping	Tax Duty shipping	N/A
Cardholder Authentication	authenticationIndicator cardholderAuthenticationValue	authenticationIndicator cardholderAuthenticationValue	N/A	N/A	N/A	N/A
Advanced Fraud Detection Suite™ (AFDS)	customerIP Required only when the merchant is using the AFDS IP blocking tool)	customerIP (required only when the merchant is using the AFDS IP blocking tool)	N/A	N/A	N/A	N/A

† These fields can support either a straight numeric value or line item details

Note

For Prior Authorization and Capture transactions, if you submitted line item information with the original transaction, you can submit adjusted information if the transaction changed. If you submit no adjusted line item information, the information submitted with the original transaction applies.

APPENDIX B

Request and Response Example

This section shows an example transaction request and response. Note that not all elements are illustrated in these examples. You can find more examples in the Developer Center at http://developer.authorize.net.

createTransactionRequest

```
<?xml version="1.0" encoding="utf-8"?>
<createTransactionRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd">
  <merchantAuthentication>
    <name>hcwt42</name>
    <transactionKey>123abc</transactionKey>
  </merchantAuthentication>
  <refId>123456</refId>
  <transactionRequest>
    <transactionType>authOnlyTransaction</transactionType>
    <amount>5</amount>
    <payment>
      <creditCard>
        <cardNumber>5424000000000015</cardNumber>
        <expirationDate>1102</expirationDate>
        <cardCode>999</cardCode>
      </creditCard>
    </payment>
    <lineItems>
      <lineItem>
        <itemId>1</itemId>
        <name>vase</name>
        <description>Cannes logo </description>
        <quantity>18</quantity>
        <unitPrice>45.00</unitPrice>
      </lineItem>
    </lineItems>
    <tax>
      <amount>4.26</amount>
      <name>level2 tax name</name>
```

```
  <description>level2 tax</description>
</tax>
<duty>
  <amount>8.55</amount>
  <name>duty name</name>
  <description>duty description</description>
</duty>
<shipping>
  <amount>4.26</amount>
  <name>level2 tax name</name>
  <description>level2 tax</description>
</shipping>
<poNumber>456654</poNumber>
<customer>
  <id>18</id>
  <email>nobody@example.com</email>
</customer>
<billTo>
  <firstName>Ellen</firstName>
  <lastName>Johnson</lastName>
  <company>Souveniropolis</company>
  <address>14 Main Street</address>
  <city>Pecan Springs</city>
  <state>TX</state>
  <zip>44628</zip>
  <country>USA</country>
</billTo>
<shipTo>
  <firstName>China</firstName>
  <lastName>Bayles</lastName>
  <company>Thyme for Tea</company>
  <address>12 Main Street</address>
  <city>Pecan Springs</city>
  <state>TX</state>
  <zip>44628</zip>
  <country>USA</country>
</shipTo>
<customerIP>192.168.1.1</customerIP>
<transactionSettings>
  <setting>
    <settingName>allowPartialAuth</settingName>
    <settingValue>false</settingValue>
  </setting>
  <setting>
    <settingName>duplicateWindow</settingName>
    <settingValue>0</settingValue>
  </setting>
  <setting>
    <settingName>Customer</settingName>
    <settingValue>false</settingValue>
  </setting>
  <setting>
    <settingName>recurringBilling</settingName>
```

```
        <settingValue>false</settingValue>
      </setting>
      <setting>
        <settingName>testRequest</settingName>
        <settingValue>false</settingValue>
      </setting>
    </transactionSettings>
    <userFields>
      <userField>
        <name>MerchantDefinedFieldName1</name>
        <value>MerchantDefinedFieldValue1</value>
      </userField>
      <userField>
        <name>favorite_color</name>
        <value>blue</value>
      </userField>
    </userFields>
  </transactionRequest>
</createTransactionRequest>
```

Encrypted Mobile Card Reader Track Data

To use encrypted track data, insert the following in the <payment> element of **createTransactionRequest**:

```
<encryptedTrackData>
  <FormOfPayment>
    <Value>
      <Encoding>Hex</Encoding>
      <EncryptionAlgorithm>TDES</EncryptionAlgorithm>
      <Scheme>
        <DUKPT>
          <Operation>DECRYPT</Operation>
          <Mode>
            <Data>1</Data>
          <Mode>
          <DeviceInfo>
            <Description>
             4649443D4944544543482E556E694D61672E416E64726F69642E53646B
76315e536f6d655442444b6579313d736f6d656f7468657276616c756
            </Description>
          </DeviceInfo>
          <EncryptedData>
            <Value>
             02f300801f342600039b252a343237352a2a2a2a2a2a2a353637355e
332f54455354205e2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a3f2a3b343
237352a2a2a2a2a2a2a2a353637353d2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a3f2a
521db2603bfe6169fc371211161c4ad6edd7294a8352af1ba8b388c527d8d6335286413
ad67521b8be085da998cef7ae3621b0b72eecd6d61953a4a268e02a8cdff3d365216df7
3646b326d8dac369c11a2a3a4a9336addc4a15ae5d8843e0163bae895b9b4df3253439b
4dd885363ad108604ea04f2e4fac701a5a0e65c54e1301a5ed7706eb887629949010000
00400015ac1e03
            </Value>
          </EncryptedData>
        </DUKPT>
      </Scheme>
    </Value>
  </FormOfPayment>
</encryptedTrackData>
```

createTransactionResponse

```
<?xml version="1.0" encoding="utf-8"?>
<createTransactionResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetA
piSchema.xsd">
  <refId>123456</refId>
```

```
  <messages>
    <resultCode>Ok</resultCode>
    <message>
      <code>I00001</code>
      <text>Successful.</text>
    </message>
  </messages>
  <transactionResponse>
    <responseCode>1</responseCode>
    <authCode>UGELQC</authCode>
    <avsResultCode>E</avsResultCode>
    <cavvResultCode />
    <transId>2148061808</transId>
    <refTransID />
    <transHash>0B428D8A928AAC61121AF2F6EAC5FF3F</transHash>
    <testRequest>0</testRequest>
    <accountNumber>XXXX0015</accountNumber>
    <accountType>MasterCard</accountType>
    <message>
      <code>1</code>
      <description>This transaction has been approved.</description>
    </message>
    <userFields>
      <userField>
        <name>MerchantDefinedFieldName1</name>
        <value>MerchantDefinedFieldValue1</value>
      </userField>
      <userField>
        <name>favorite_color</name>
        <value>lavender</value>
      </userField>
    </userFields>
  </transactionResponse>
</createTransactionResponse>
```

APPENDIX C

Information and Error Messages

The following table lists common information and error message codes and text. Messages that being with an I are information messages. Messages that begin with an E are error messages.

Table 13 Information and Error Messages

Code	Text	Description
I00001	Successful	The record was processed successfully
I00003	The record has already been deleted	The record has already been deleted
I00005	The mobile device has been submitted for approval by the account administrator	
I00006	The mobile device is approved and ready for use	
E00001	An error occurred during processing. Try again.	An unexpected system error occurred while processing this request.
E00002	The content-type specified is not supported.	The only supported content-types are text/xml and application/xml.
E00003	An error occurred while parsing the XML request.	This is the result of an XML parser error.
E00004	The name of the requested API method is invalid.	The name of the root node of the XML request is the API method being called. It is not valid.
E00005	The merchantAuthentication.transactionKey is invalid or not present.	Merchant authentication requires a valid value for transaction key.
E00006	The merchantAuthentication.name is invalid or not present.	Merchant authentication requires a valid value for name.
E00007	User authentication failed due to invalid authentication values.	The name/and or transaction key is invalid.
E00008	User authentication failed. The payment gateway account or user is inactive.	The payment gateway or user account is not currently active.
E00009	The payment gateway account is in Test Mode. The request cannot be processed.	The requested API method cannot be executed while the payment gateway account is in Test Mode.
E00010	User authentication failed. You do not have the appropriate permissions.	The user does not have permission to call the API.
E00011	Access denied. You do not have the appropriate permissions.	The user does not have permission to call the API method.

Table 13 Information and Error Messages (Continued)

Code	Text	Description
E00012	A duplicate subscription already exists.	A duplicate of the subscription was already submitted. The duplicate check looks at several fields including payment information, billing information and, specifically for subscriptions, Start Date, Interval and Unit.
E00013	The field is invalid.	One of the field values is not valid.
E00014	A required field is not present.	One of the required fields was not present.
E00015	The field length is invalid.	One of the fields has an invalid length.
E00016	The field type is invalid.	The field type is not valid.
E00017	The startDate cannot occur in the past.	The subscription start date cannot occur before the subscription submission date. **Note** Validation is performed against local server time, which is Mountain Time.
E00018	The credit card expires before the subscription startDate.	The credit card is not valid as of the start date of the subscription.
E00019	The customer taxId or driversLicense information is required.	The customer tax ID or driver's license information (driver's license number, driver's license state, driver's license DOB) is required for the subscription.
E00020	The payment gateway account is not enabled for eCheck.Net subscriptions.	This payment gateway account is not set up to process eCheck.Net subscriptions.
E00021	The payment gateway account is not enabled for credit card subscriptions.	This payment gateway account is not set up to process credit card subscriptions.
E00022	The interval length cannot exceed 365 days or 12 months.	The interval length must be 7 to 365 days or 1 to 12 months.
E00024	The trialOccurrences is required when trialAmount is specified.	The number of trial occurrences cannot be zero if a valid trial amount is submitted.
E00025	Automated Recurring Billing is not enabled.	The payment gateway account is not enabled for Automated Recurring Billing.
E00026	Both trialAmount and trialOccurrences are required.	If either a trial amount or number of trial occurrences is specified then values for both must be submitted.
E00027	The test transaction was unsuccessful.	An approval was not returned for the test transaction.
E00028	The trialOccurrences must be less than totalOccurrences.	The number of trial occurrences specified must be less than the number of total occurrences specified.
E00029	Payment information is required.	Payment information is required when creating a subscription.
E00030	A paymentSchedule is required.	A payment schedule is required when creating a subscription.

Table 13 Information and Error Messages (Continued)

Code	Text	Description
E00031	The amount is required.	The subscription amount is required when creating a subscription.
E00032	The startDate is required.	The subscription start date is required to create a subscription.
E00033	The subscription Start Date cannot be changed.	Once a subscription is created the Start Date cannot be changed.
E00034	The interval information cannot be changed.	Once a subscription is created the subscription interval cannot be changed.
E00035	The subscription cannot be found.	The subscription ID for this request is not valid for this merchant.
E00036	The payment type cannot be changed.	Changing the subscription payment type between credit card and eCheck.Net is not currently supported.
E00037	The subscription cannot be updated.	Subscriptions that are expired, canceled or terminated cannot be updated.
E00038	The subscription cannot be canceled.	Subscriptions that are expired or terminated cannot be canceled.
E00045	The root node does not reference a valid XML namespace.	An error exists in the XML namespace. This error is similar to E00003.
E00054	The mobile device is not registered with this merchant account.	
E00055	The mobile device is pending approval by the account administrator.	
E00056	The mobile device has been disabled for use with this account.	
E00057	The user does not have permissions to submit requests from a mobile device.	
E00058	The merchant has met or exceeded the number of pending mobile devices permitted for this account.	An account is limited to 100 pending mobile devices. You need to enable, disable, or delete some devices.
E00059	The authentication type is not allowed for this method call.	
E00098	Customer Profile ID or Shipping Profile ID not found.	Search for shipping profile using customer profile id and shipping profile id did not find any records.
E00099	Customer profile creation failed. This transaction ID is invalid.	Customer profile creation failed. This transaction ID is invalid.
E000100	Customer profile creation failed. This transaction type does not support profile creation.	Customer profile creation failed. This transaction type does not support profile creation.
E000101	Customer profile creation failed.	Error creating a customer payment profile from transaction.

Table 13 Information and Error Messages (Continued)

Code	Text	Description
E000102	Customer Info is missing.	Error creating a customer profile from transaction.

APPENDIX D

Track Data

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

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

The longitudinal redundancy checks (LRCs) must be calculated for the data read from the track and compared to the LRCs read from the track. The track data is assumed to be read without errors when no character parity errors are detected and the calculated and read LRCs match.

The starting sentinel, ending sentinel, and LRC are discarded.

The character codes read from the magnetic stripe must be converted from the encoded character set to the set used for the authorization request message. The characters encoded on Track 1 are 6-bit plus parity codes, and the characters encoded on Track 2 are 4-bit plus parity codes, with the character set used for the request message defined as 7-bit plus parity code. All characters read from a track must be converted to the request message character set and transmitted as part of the request. The converted track data can not be modified by adding or deleting non-framing characters and must be a one-for-one representation of the characters read from the track.

Note

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

Track 1 Data

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

The Track 1 data read from the cardholder's card is checked for parity and LRC errors and then converted from the 6-bit characters encoded on the card to 7-bit characters as defined in ANSI X3.4.

As part of the conversion, the terminal must remove the framing characters (start sentinel, end sentinel, and LRC characters). The separators must be converted to either an ASCII "^" (HEX 5E) or ASCII <US> (HEX 1F) characters. The entire UNALTERED track (excluding framing characters) must be provided in the authorization request message or an error condition results.

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

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

Legend:

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

Track 2 Data

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

The Track 2 data read from the cardholder's card is checked for parity and LRC errors and then converted from the 4-bit characters encoded on the card to 7-bit characters as defined in ANSI X3.4. As part of the conversion, the terminal must remove the start sentinel, end sentinel, and LRC characters. The separators must be converted to either an ASCII "=" (HEX 3D) or ASCII "D" (HEX 44) characters. The entire UNALTERED track (excluding framing characters) must be provided in the authorization request message or an error message is generated.

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

SS	PAN	FS	DATE	SVC CD	DISCRETIONARY DATA	ES	LRC

Legend:

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

APPENDIX E

Supported Card Readers

Only card readers that have been injected with the Authorize.Net security key are supported. The card readers below only work with certain devices and carriers. Before purchasing, we recommend that you check to make sure your device and carrier are compatible. The following links can be used to find this information.

How to Order

Approved card readers may be purchased at the following location:

http://www.authorize.net/solutions/merchantsolutions/merchantservices/mobileapp/#secure_card_readers