

This document and the API that it describes are deprecated.

Authorize.Net's legacy name-value-pair API is still supported, however it will not be updated, except for critical security updates. To learn when this deprecated API will reach its end of life, and for information on upgrading to our latest API, read the [Upgrade Guide](#). You can find the full Authorize.Net API documentation at our [Developer Center](#).

Server Integration Method (SIM) Card-Not-Present Transactions

Developer Guide

September 2017

Note: The SIM API is deprecated. It will only receive critical security updates. We recommend [Accept Hosted](#) instead of SIM for new integrations.

Authorize.Net[®]

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™

Automated Recurring Billing™

eCheck.Net®

The logo for Authorize.Net, featuring the word "Authorize.Net" in a blue, serif font. The "A" is significantly larger than the other letters, and the ".Net" is in a smaller size. A registered trademark symbol (®) is located at the top right of the "t".

Contents

Recent Revisions to This Document 8

About This Guide 9

Audience and Purpose 9

Conventions 9

 Note, Important, and Warning Statements 9

 Text and Command Conventions 10

Developer Support 10

10

Chapter 1 Introduction 11

Other Integration Methods 11

 AIM 11

 DPM 12

SIM Minimum Requirements 12

Managing Integration Settings 13

Features of SIM 13

eCheck.Net 15

Visa Checkout 15

Payment Processors 16

 North American Payment Processors 16

 European Payment Processors 18

 Asia-Pacific Processors 18

 EVOSnap 19

 Accepted Authorization/Settlement Currencies 19

 Accepted Billing Currencies 19

 Accepted Card Types 19

 Unsupported Services 19

 EVOSnap Supported Services 20

Software Development Kits 23

Chapter 2	Transaction Data Requirements	24
	Credit Card Transaction Types	24
	Authorization and Capture	24
	Authorization Only	25
	Prior Authorization and Capture	25
	Capture Only	26
	Credit	26
	Void	26
	Partial Authorization Transactions	27
	Using the Merchant Interface	27

Chapter 3	Submitting Transactions	28
	Transaction Post Location	28
	Generating the Unique Transaction Fingerprint	28
	Custom Transaction Fingerprint Code	29
	The Signature Key	30
	Requesting the Secure Hosted Payment Form	31
	Configuring the Hosted Payment Form Fields	35
	EVO Billing and Shipping Fields	41
	Configuring the Appearance of the Hosted Payment Form	43
	Placement of Custom Headers and Footers	47
	Adding a Cancel Link	48
	Logos and Background Images for the Hosted Payment Form	49
	Image Requirements and Guidelines	50
	Merchant-Defined Fields	50

Chapter 4	Receipt Options	52
	Using the Hosted Receipt Page	52
	Receipt Link URL(s)	53
	Receipt Method	53
	Customizing the Receipt Page	57
	Relay Response	59
	Whitelisting	61
	Tips for Using Relay Response	61
	Email Receipt	62

Chapter 5	Additional API Fields	64
	Transaction Information	64
	Itemized Order Information	65

Additional Customer Information	66
x_customer_ip	66

Chapter 6	Transaction Response	68
	Fields in the Payment Gateway Response	68
	Using the HMAC SHA-512 Hash Feature	73
	Response for Duplicate Transactions	75
	SIM Relay Response	76
	SIM Transaction Response Versions	76
	Version 3.0	76
	Version 3.1	76
	Upgrading the Transaction Version	76
	Response Code Details	77
	Response Codes	77
	Response Reason Codes and Response Reason Text	77
	Response Example for Partial Authorization Transactions	89

Chapter 7	Testing Your Integration	90
------------------	---------------------------------	-----------

Appendix A	Fields by Transaction Type	91
	Minimum Required Fields	91
	Required Fields for Advanced SIM Features	92
	Best Practice Fields	92

Appendix B	Alphabetized List of API Fields	94
-------------------	--	-----------

Appendix C	Direct Post Method (DPM)	108
	Differences From SIM	108
	Relay Response	109
	Conceptual Overview	110
	Address and Card Code Verification	111

Appendix D	Visa Checkout	112
	Enabling Visa Checkout for Your Account	112
	Visa Checkout API Fields	113
	Browser Support	114

Index 115

Recent Revisions to This Document

The following table lists the most recent revisions to this guide.

Release	Changes
April 2017	Added the section " Using the HMAC SHA-512 Hash Feature ," page 73.
October 2016	<p>Deprecated the DPM method in favor of our newer method, Accept.js, a JavaScript payment form. Accept.js is a modern implementation that doesn't require a post-back. With Accept.js, developers have control over the user experience without sending PCI data through their servers. Accept.js can be used like any other payment type in the Authorize.Net API.</p> <p>We strongly encourage new development using Accept.js instead of DPM, and developers with existing DPM implementations should consider moving to Accept.js.</p>
January 2015	Updated " Testing Your Integration ," page 90 by removing outdated information and replacing it with a link to the Testing Guide .
December 2015	This revision contains only editorial changes and no technical updates.
November 2015	Updated the sandbox URL to which you should send test transactions to https://test.authorize.net/gateway/transact.dll . For more information about sending test transactions to the sandbox server, see " Testing Your Integration ," page 90.

About This Guide

Audience and Purpose

This guide is intended for developers. It describes the web development necessary in order to use the Server Integration Method (SIM) API to connect an e-commerce web site or other application to the Authorize.Net Payment Gateway.

Conventions

Note, Important, and Warning Statements



Note

A *Note* contains helpful suggestions or references to material not contained in the document.



Important

An *Important* statement contains information essential to successfully completing a task or learning a concept.



Warning

A *Warning* contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Text and Command Conventions

Convention	Usage
bold	<ul style="list-style-type: none"> Field and service names in text; for example: Include the ics_applications field. Items that you are instructed to act upon; for example: Click Save.
<i>italic</i>	<ul style="list-style-type: none"> Filenames and pathnames. For example: Add the filter definition and mapping to your <i>web.xml</i> file. Placeholder variables for which you supply particular values.
monospace	<ul style="list-style-type: none"> XML elements. Code examples and samples. Text that you enter in an API environment; for example: Set the davService_run field to <code>true</code>.

Developer Support

The following resources can 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.

Introduction

The Server Integration Method (SIM) is a hosted payment processing solution that handles all of the steps in processing a transaction, including:

- Collecting customer payment information through a secure, hosted form
- Generating a receipt to the customer
- Securely transmitting to the payment processing networks for settlement
- Funding of proceeds to the merchant's bank account
- Securely storing cardholder information

The security of a SIM transaction is ensured by the unique digital signature or “fingerprint” that is sent with each transaction. Authorize.Net uses this fingerprint to authenticate both the merchant and the transaction.

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

The SIM API consists of required and optional form fields that can be submitted to the payment gateway for real-time transaction processing. The API includes fields for requesting the payment gateway's secure hosted payment form, which can be customized to reflect the look and feel of the merchant's web site.

Other Integration Methods

AIM

The Advanced Integration Method (AIM) is designed for merchants who need a highly customizable payment form (for example, complete control of look and feel and the ability to keep the customer on their web site during the entire checkout process) or for

merchants who are integrating a standalone business application. For more information about AIM, see the *AIM Developer Guide*:

<http://developer.authorize.net/guides/AIM/>

DPM

The Direct Post Method (DPM) is a hosted payment option that enables the developer to customize while still relying on Authorize.Net for help with PCI compliance. DPM uses a unique fingerprint to authenticate transactions, so developers customize a secure hosted payment form without needing an SSL/TLS certificate. The Authorize.Net Payment Gateway handles all the steps in the secure transaction process—payment data collection, data submission, and the response to the customer—while keeping Authorize.Net virtually transparent. For more information on implementing DPM, see "Direct Post Method (DPM)," page 108.



Important

We now offer [Accept.js](#), a JavaScript replacement for Direct Post Method (DPM). Accept.js is a modern implementation that doesn't require a post-back. With Accept.js, developers have control over the user experience without sending PCI data through their servers. Accept.js can be used like any other payment type in the [Authorize.Net API](#).

We strongly encourage new development using Accept.js instead of DPM, and developers with existing DPM implementations should consider moving to Accept.js.

SIM Minimum Requirements

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

- The merchant must have a merchant bank account that allows Internet transactions.
- The merchant must have an Authorize.Net Payment Gateway account.
- The merchant's web site must be capable of performing an HTML Form POST to request the secure payment gateway hosted payment form.
- The merchant's web site or hosting provider must have server scripting or CGI capabilities such as ASP Classic, Cold Fusion, PHP, or Perl.

- The merchant must be able to store payment gateway account data securely (for example, API Login ID or Transaction Key).

**Note**

Merchants should avoid storing any type of sensitive cardholder information. However, if a merchant or third party must store sensitive customer business or payment information, compliance with industry standard storage requirements is required. See [Understanding PCI Compliance](#).

Managing Integration Settings

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

- Included in the transaction request per transaction by using the application programming interface (API) as described in this guide
- Configured in the Merchant Interface and applied to all transactions

**Important**

The Merchant Interface at <https://secure.authorize.net> is a secure web site where merchants can manage their payment gateway account settings, including their web site integration settings. We recommend that you review the *Merchant Integration Guide* for information on managing the payment gateway integration using the Merchant Interface:

<http://www.authorize.net/support/merchant/>

Transaction settings submitted in the transaction request 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, you should review the integration settings that can be configured in the Merchant Interface with the merchant and determine which integration settings can be posted per transaction, and which should be configured in the Merchant Interface. See [Appendix A, "Fields by Transaction Type," on page 91](#) for a list of fields the payment gateway recommends be submitted per transaction.

Features of SIM

In addition to basic transaction processing, SIM provides merchants with several features for configuring transaction security options and further customizing their customer

checkout experience. These features are listed in [Table 1](#). Take a few moments to discuss them with your merchant and select features to include in their integration.

Table 1 Features of SIM

Feature	Description	Requirements
Address Verification Service (AVS) filter	This feature enables merchants to compare the billing address that the customer submits for the transaction to 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, the merchant must require that the customer fill out the Address and ZIP Code fields on the payment gateway hosted payment form. For more information about AVS, see the <i>Merchant Integration Guide</i> : http://www.authorize.net/support/merchant
Card Code Verification (CCV) filter	This feature enables merchants to compare the card code that the customer submits 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, the merchant must require that the customer fill out the Card Code field on the payment gateway hosted payment form. For more information CCV, see the <i>Merchant Integration Guide</i> : 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 email, in the Transaction Details for the transaction, and in QuickBooks download reports in the Merchant Interface.	To implement itemized order information, you must submit line item fields per transaction. See "Itemized Order Information," page 65 , for details.
Receipt page	This feature enables merchants to customize the payment gateway hosted receipt page that is displayed to the customer at the completion of a transaction. This page can include a hyperlink back to the merchant's web site.	To configure the payment gateway hosted receipt page, you must either configure settings in the Receipt Page section of the Settings menu in the Merchant Interface or submit them per transaction. See "Receipt Options," page 52 , for details.

Table 1 Features of SIM (Continued)

Feature	Description	Requirements
Email receipt	This feature enables merchants to request that the payment gateway send an automatic email receipt to their customers.	To configure the payment gateway email receipt, the merchant must require that the customer fill out the email address on the hosted payment form. You must also configure the Email Receipts section of the Settings menu in the Merchant Interface or submit these settings per transaction. See " Receipt Options ," page 52, for details.
Relay response	This feature enables merchants to display a more customized receipt page that is generated on the merchant's web server and relayed by the payment gateway to the customer's browser.	To configure relay response, you must configure the settings in the Relay Response section of the Settings menu in the Merchant Interface or submit them per transaction. See " Relay Response ," page 59, for details.

eCheck.Net

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

<http://www.authorize.net/support/eCheck.pdf>

Visa Checkout

Visa Checkout can be used as a payment option on the payment form. For more information, see "[Visa Checkout](#)," page 112.

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, card types, and currencies.

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

Payment Processor	Accepted Card Types	Accepted Currencies
Chase Paymentech Tampa	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD) Canadian Dollar (CAD)
Elavon	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD) Canadian Dollar (CAD)
EVO Payments	<ul style="list-style-type: none"> ■ American Express ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD)
First Data Merchant Services (FDMS) Omaha, Nashville, and EFSNet	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD) Canadian Dollar (CAD)

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

Payment Processor	Accepted Card Types	Accepted Currencies
Global Payments	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD) Canadian Dollar (CAD)
Heartland Payment Systems	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD)
TSYS Acquiring Solutions	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD)
WorldPay Atlanta	<ul style="list-style-type: none"> ■ American Express ■ Diners Club ■ Discover ■ JCB ■ Mastercard ■ Visa 	United States Dollar (USD)

European Payment Processors

Authorize.Net supports the following European payment processors, card types, and currencies.

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

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

Asia-Pacific 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	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	Australian Dollar (AUD) New Zealand Dollar (NZD) United States Dollar (USD)
Westpac	<ul style="list-style-type: none"> ■ Mastercard ■ Visa 	Australian Dollar (AUD)

EVOSnap

There are multiple EVOSnap processing platforms. If you use the U.S. Dollar (USD), you are assigned to EVOSnap U.S. If you use any other currencies, you are assigned to EVOSnap International.

Accepted Authorization/Settlement Currencies

- USD—U.S. Dollar
- CAD—Canada Dollar
- CHF—Swiss Franc
- DKK—Danish Krone
- EUR—Euro
- GBP—British Pound
- NOK—Norway Krone
- PLN—Poland złoty (MasterCard Only)
- SEK—Sweden Krone
- ZAR—South African Rand

Accepted Billing Currencies

- USD—U.S. Dollar
- AUD—Australian Dollar
- GBP—British Pound

Accepted Card Types

- Visa
- MasterCard
- American Express
- JCB
- Diners Club—Supported for U.S. services only.

Unsupported Services

Apple Pay and soft descriptors are not supported by EVOSnap.

EVOSnap Supported Services

U.S. Services

Table 5 Authorize.Net Services Supported by EVOSnap U.S.

Service	E-Commerce	MOTO	Retail
Transaction Types			
Authorization only	X	X	X
Authorization and capture (sale)	X	X	X
Capture	X	X	X
Void	X	X	X
Credit (refund)	X	X	X
Features			
AVS	X	X	X
CVV2/CVC2/CID	X	X	X
3DS	X		
Purchase card—level 2	X	X	X
Partial Approvals (Partial Authorization)	X	X	X
Supported card types:	X	X	X
<ul style="list-style-type: none"> ■ Visa ■ Mastercard ■ American Express ■ Discover ■ JCB ■ Diners Club 			

Duplication Rules

EVO platform always checks for duplicate transactions based on:

- Same Terminal ID
- Same Card Number
- Same Dollar Amount

Duplicates are flagged when they occur within an hour of each other.

Magstripe

Track 2 data is supported only for card-present transactions.

Level 2 Support

PO# is required when any level 2 data is submitted. Level 2 data includes tax, duty, and freight information.

Billing Address

When any billing fields are submitted, all must be submitted.

- First name
- Last name
- Address
- City
- State/province (only required if country is US or Canada)
- Country
- ZIP/postal code

Other Field Requirements

The **employeeid** field is required; however, if a value is not passed with the field, Authorize.Net sends a default value of 0000 to the processor.

Consolidated Accounts

The Consolidated Accounts feature is not supported on the EVOSnap platform. Multiple market types require multiple accounts.

Automated Recurring Billing

Merchants using Automated Recurring Billing must be approved by their merchant service provider, also known as their acquirer.

International Services

Table 6 Authorize.Net Services Supported by EVOSnap International

Service	MOTO	E-Commerce
Authorize	X	X
Authorize and Capture	X	X
Capture	X	X
Void	X	X
Credit	X	X
AVS—Visa and American Express only.	X	X
CVV2/CVC2/CID	X	X
3DS		X

Not Supported

- Retail
- Level 2 data
- Soft descriptors
- Partial authorization
- Consolidated accounts (MOTO/E-Commerce)—separate accounts are required.
- Automated recurring billing and customer information manager

CVV

EVOSnap requires CVV for all international transactions. CVV must be enabled in the Authorize.Net merchant interface's Virtual Terminal settings.

To enable CVV:

-
- Step 1** Navigate to the Authorize.Net merchant interface.
 - Step 2** Choose **Accounts > Settings > Transaction Format Settings > Virtual Terminal**.
 - Step 3** Check the **View/Edit** box for **Card Code**.
 - Step 4** Click **Submit**.
-

Other EVOSnap Considerations

International AVS Behavior

Transactions are declined if the submitted address data does not match. Merchants can override this behavior on a per-transaction basis, if permitted by EVOSnap. Merchant

accounts are configured to either use or not use AVS processing when they are boarded. If the account is configured to not use AVS processing, AVS is not performed, even if the data is included. If the merchant account is configured to use AVS every transaction must include AVS data, unless the merchant is authorized by EVOSnap to override the AVS processing.

API

Customer code is required. If not present, customer code is populated with 0000. Country code must be in ISO format. For example, GBR, CHE, AUS.

Error Codes

- **RTC 350**
Description—EVOSnap: country must be a valid three-character value if specified.
Message—country must be a valid three-character value if specified.
- **RTC 351**
Description—EVOSnap: employee ID cannot be more than 6 characters in length, 4 for a retail transaction.
Message—employee ID must be 1 to %x characters in length.
Note—the %x is replaced with a 6 for E-Commerce and MOTO transaction types and 4 for retail transaction types.

Billing Information

When any billing information is submitted, all billing fields must be provided.

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

Software Development Kits

Authorize.Net offers software development kits (SDKs) that present an alternate object-oriented model, in several popular languages. To use an SDK, the merchant's transaction version must be set to 3.1. The SDK performs the core payment activities (such as error handling and parsing, network communication, and data encoding). The SDKs provide utilities to help developers build payment flows for each of the integration methods. You can download an SDK:

<http://developer.authorize.net/downloads/>

Transaction Data Requirements

The payment gateway supports several credit card transaction types for transactions submitted using SIM.

To implement SIM for a merchant's web site, you must develop an HTML Form POST to request Authorize.Net's secure payment gateway hosted payment form. This request contains required and optional merchant and transaction information.

Minimum form field requirements for posting credit card transaction requests to the payment gateway are shown in [Table 8, page 31](#).

Credit Card Transaction Types

Discuss requirements with the merchant to ensure that your integration supports them:

- 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?



Note

Some of the field requirements listed in this section are *in addition* to the minimum field requirements 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 91](#).

Authorization and Capture

Authorization and Capture (**auth_capture**) is the default transaction type in the Virtual Terminal. If no **x_type** variable is submitted with a web site transaction request, the type defaults to **auth_capture**. This type of transaction is completely automatic; the transaction is submitted to your processor for authorization and, if approved, is placed in your Unsettled Transactions list already set to capture. The transaction settles with your next

batch settlement. Settlement occurs every 24 hours, within 24 hours of the time specified in your Settings menu, under Transaction Cutoff Time.

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

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

Authorization Only

This transaction type is sent for authorization only. When an Authorization Only (**auth_only**) transaction is submitted, it is sent to your processor for authorization. If approved, the transaction is placed in your Unsettled Transactions list with a status of Authorized/Pending Capture. The authorization places the funds on hold with the customer's bank, but until the transaction is captured, the funds are not transferred. This type of transaction is not sent for settlement until you submit a Prior Authorization and Capture credit card transaction type, or you submit the transaction for capture manually in the Merchant Interface. This option can be useful when you need to make a sale but can't ship merchandise for several days; you can authorize the transaction to ensure the availability of funds, then you can capture the transaction to obtain the funds when you ship.

Authorization Only transactions are listed as unsettled for 30 days. After 30 days, transaction status changes to Expired, and the funds are NOT transferred. To capture a transaction, you can manually log on to your Authorize.Net interface and view your unsettled transactions list. From there, you can use the group capture filter to capture multiple transactions at once, or click the individual transaction ID of the transaction you wish to capture, and the next screen will provide a Capture button. From a web site or billing application, you can submit the **x_type** variable with a value of `Prior_Auth_Capture` to capture the transaction.

The unique field requirement for an Authorization Only transaction is:

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



Note

Merchants who use SIM can configure the hosted payment form to submit either Authorization and Capture or Authorization Only transactions. Ask the merchant which of these transaction types they require.

Prior Authorization and Capture

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

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

Capture Only

Capture Only transactions are used when you already have an authorization from a bank. To use this type of transaction, you must have an authorization code from the card issuer (usually a 5- or 6-digit number). For example, if you called Visa directly and obtained an authorization over the phone, you would submit a Capture Only transaction to start the funds transfer process. You can manually submit a Capture Only transaction from your Virtual Terminal by selecting Capture Only, or from a web site or billing application by including the following variables with your transaction request:

- **x_type** (Capture_Only)
- **x_auth_code** (the 5- or 6-digit code provided by the card issuer)

Credit

This transaction type issues a refund to a customer for a transaction that was originally processed and successfully settled through the payment gateway. Credit transactions can be submitted for 120 days after the original authorization was obtained. To issue a credit for a transaction not submitted through the payment gateway, or for a transaction submitted more than 120 days before, you must apply for Expanded Credit Capability (you can find the request form at <http://www.authorize.net/files/ecc.pdf>). Credits can be manually processed through the Virtual Terminal or can be submitted from a web site or billing application.

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

Void

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

You can manually void transactions from the Unsettled Transactions screen of the Merchant Interface. Use the Group Void filter toward the top of your screen to void multiple transactions at once, or click the individual transaction ID of the transaction you would like to void; the next screen will provide a Void button.

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

Partial Authorization Transactions

A partial authorization, or *split tender*, order is one in which two or more transactions are used to cover the total amount of the order.

The merchant must either select the Partial Authorization option in the Account settings of the Merchant Interface or send `x_allow_partial_auth=true` with each 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 authorized for a partial amount, a split tender ID is generated and returned in the response. This ID must be passed back with each of the remaining transactions of the group, using `x_split_tender_id=<value>`. If you include both a split tender ID and a transaction ID on the same request, an error results.

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

The following fields are returned in the relay response data sent to the merchant's URL. The data they correspond to are in all prepaid card responses.

- `x_prepaid_requested_amount`—the amount requested.
- `x_split_tender_id`—the split-tender ID provided when the first partial authorization transaction was issued.
- `x_split_tender_status`—indicates whether or not the transaction is complete.
- `x_card_type`—the card type.



Note

The payment processor EVO does not support partial authorizations.

Using the Merchant Interface

Using the Merchant Interface, merchants can capture Authorize Only transactions, void transactions, and issue refunds. You can also manage these transaction types automatically through the API. 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.

Submitting Transactions

Transaction Post Location

The merchant's web site should post transaction requests through an HTML Form POST to the following payment gateway URL:

<https://secure2.authorize.net/gateway/transact.dll>

Legacy but supported URL:

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



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

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



Do not submit transactions to an IP address. Use the URLs shown above.

Generating the Unique Transaction Fingerprint



Authorize.Net recommends the HMAC-SHA512 hashing algorithm for constructing the fingerprint. HMAC-MD5 is still supported but not recommended.

Transaction authentication for SIM is a transaction fingerprint, or a hash of merchant- and transaction-specific information using the HMAC-SHA512 hashing algorithm. The HMAC-SHA512 algorithm is used only for generating the unique transaction fingerprint. The

transaction fingerprint must be generated for each transaction by a server-side script on the merchant's web server and inserted into the transaction request. The payment gateway uses the same mutually exclusive merchant information to decrypt the transaction fingerprint and authenticate the transaction.

Custom Transaction Fingerprint Code

The following table contains API field requirements for constructing the fingerprint code. Use the following syntax when you insert the form fields into the transaction request:

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

Table 7 Field Requirements for the Transaction Fingerprint

Field Name	Description
x_fp_hash	<p>Value: The unique transaction fingerprint.</p> <p>Notes: The fingerprint is generated using the HMAC-SHA512 hashing algorithm on the following field values:</p> <ul style="list-style-type: none"> ■ API Login ID (x_login) ■ The sequence number of the transaction (x_fp_sequence) ■ The timestamp of the sequence number creation (x_fp_timestamp) ■ Amount (x_amount) <p>Field values are concatenated and separated by the ^ character.</p>
x_fp_sequence	<p>Value: The merchant-assigned sequence number for the transaction.</p> <p>Format: Numeric.</p> <p>Notes: Merchant-assigned value, such as an invoice number or any randomly generated number.</p>
x_fp_timestamp	<p>Value: The timestamp at the time of fingerprint generation.</p> <p>Format: UTC time in seconds since January 1, 1970.</p> <p>Notes: Coordinated Universal Time (UTC) is an international atomic standard of time (sometimes referred to as GMT). Using a local time zone timestamp will cause fingerprint authentication to fail.</p> <p>If the fingerprint is more than 1 hour old or more than 15 minutes into the future, it is rejected.</p>

The transaction fingerprint that is submitted in the **x_fp_hash** field is generated using an HMAC-SHA512 hashing algorithm on the following field values:

- API login ID (**x_login**)
- Sequence number (**x_fp_sequence**)
- UTC timestamp in seconds (**x_fp_timestamp**)

**Note**

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

- Amount (**x_amount**)

**Note**

The amount used to generate the fingerprint must reflect the final amount of the transaction. To avoid any discrepancy, we strongly recommend that you generate the fingerprint at a point in the checkout process when the amount can no longer be changed.

When you generate the fingerprint script, the input values you provide must be in the field order listed above and concatenated by the caret (^) character. All trailing spaces must be removed from input values. If the fingerprint is generated using any other field order, authentication fails, and the transaction is rejected.

Example 1 Fingerprint Input Field Order

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

Note the required trailing caret (^) character. If you specify **x_currency_code**, then the value (for example, LVL) must be placed after the trailing caret.

Example 2 Fingerprint Input with Currency Code Specified

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

The Signature Key

The cryptographic key used in the HMAC calculation is the merchant's unique Signature Key, which is a value generated by the payment gateway. The merchant obtains this value from the Merchant Interface.

To generate the signature key:

- Step 1** In the merchant interface, go to **Account > Settings > Security Settings > General Security Settings > API Credential & Keys**.
- Step 2** Answer the secret question.
- Step 3** Select **New Signature Key**. Your signature key is displayed as a string.
- Step 4** Click Copy to Clipboard.

Example 3 Generating the Transaction Fingerprint

```
Fingerprint = HMAC-SHA512
("authnettest^789^67897654^10.50^", "72207A5E14B41DD15E473510AF35F5F0972FD
6E5D421532C36B47A126F30512CA230F0F73C45D5FAC07D8A8C3265AE20B220FAB37B6674
91FCD5D1C11B8E0F5B")
```

**Note**

The signature key is provided as a HEX string. If the HMAC function that you are using requires a text string, it must be converted.

Requesting the Secure Hosted Payment Form

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

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

Table 8 Minimum Fields Required for Requesting the Hosted Payment Form

Field Name	Description
x_login	<p>Value: The merchant's unique API login ID.</p> <p>Format: 20-character maximum.</p> <p>Notes: The merchant API login ID is provided in the Merchant Interface and must be stored securely.</p> <p>The API login ID and transaction fingerprint together provide the merchant authentication required for access to the payment gateway.</p> <p>See the <i>Merchant Integration Guide</i> for more information: http://www.authorize.net/support/merchant</p>
x_type	<p>Value: The type of credit card transaction.</p> <p>Format: AUTH_CAPTURE (default), AUTH_ONLY.</p> <p>Notes: If the value submitted does not match a supported value, the transaction is rejected. If this field is not submitted, or the value is blank, the payment gateway processes the transaction as an AUTH_CAPTURE.</p>
x_amount	<p>Value: The amount of the transaction.</p>
x_show_form	<p>Value: The payment form request.</p> <p>Format: PAYMENT_FORM.</p> <p>Notes: This field indicates that the merchant would like to use the payment gateway hosted payment form to collect payment data.</p>

Table 8 Minimum Fields Required for Requesting the Hosted Payment Form

Field Name	Description
x_relay_response	Value: Indicates whether a relay response is desired. For more information, see "Relay Response," page 59. Format: TRUE, FALSE

**Note**

European payment processors require additional fields. For more information, see ["Billing Information," page 36.](#)

[Example 4](#) shows the minimum requirements for requesting the hosted payment form. It also shows how to produce a button ([Figure 1](#)) that is displayed to the customer upon checkout. When the customer clicks the button, the secure hosted payment form ([Figure 2](#)) is displayed in the customer's browser.

The code also shows that the fingerprint hash function inserts the required input fields into the HTML Form POST. When the customer clicks the button, the merchant's server performs the following actions:

- 1 Generates the sequence number.
- 2 Calculates the final total amount of the transaction.
- 3 Generates the transaction fingerprint (InsertFP).
- 4 Directs the customer's web browser to the hosted payment form.

**Note**

The code included in this document uses simulated field values in an ASP scripting environment. Code varies based on web programming language, so we do not recommend that you copy and paste it but rather use it as a guide.

Example 4 Submitting a Request for the Hosted Payment Form

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

Even though the **x_version** field included in the example above is not technically a minimum requirement for submitting a transaction, we recommend that you submit this

field per transaction, especially if you are using Relay Response. For more information, see ["Additional API Fields," page 64](#).

**Note**

We do not recommend using frames with the hosted payment form. The hosted payment form is secure; however, the frame determines the presence of the lock icon in the user's browser, so it will not appear.

Figure 1 **Payment Form Button**



Figure 2 The Secure Hosted Payment Form

Header 2

Payment Form Header

Order Information * Required Fields

Invoice Number:

Description:

Total: US \$4.50

Payment Information

Pay by Credit Card Bank Account (USA only)



Card Number: * (enter number without spaces or dashes)

Expiration Date: * (mmyy)

Billing Information

Customer ID: Customer

First Name: Last Name:

Company:

Address:

City:

State/Province: Zip/Postal Code:

Country:

Email:

Phone:

Fax:

Shipping Information

Copy Billing Information to Shipping Information

First Name: Last Name:

Company:

Address:

City:

State/Province: Zip/Postal Code:

Country:

Payment Form Footer

[Cancel URL text](#)

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

- Amount
- Credit Card Number
- Expiration Date

Configuring the Hosted Payment Form Fields

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



Important

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

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



Note

These field attributes dictate only what is displayed on the hosted payment form. Any fields that are submitted with the HTML Form POST but that do not have attributes configured in the Merchant Interface are still submitted with the transaction to the payment gateway. Merchants requesting a Relay Response can therefore receive transaction or order information that is not necessary for the customer to view or submit. For more information on Relay Response, see "Receipt Options," page 52.

For information on configuring payment form fields and attributes in the Merchant Interface, see the *Merchant Integration Guide*:

<http://www.authorize.net/support/merchant>

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

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

Table 9 Payment Form Fields

Field Name	Description
Payment Information	
x_recurring_billing	<p>Value: The recurring billing status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Marker used by merchant account providers to identify transactions that originate from merchant-hosted recurring billing applications. This value is not affiliated with automated recurring billing.</p>
x_currency_code	<p>Optional.</p> <p>Value: AUD, USD, CAD, EUR, GBP or NZD.</p> <p>Format: 3-character string.</p> <p>Notes: If you do not submit this field, the payment gateway uses the currency selected by the merchant's payment processor. Setting this field to a currency that is not supported by the payment processor results in an error.</p>
Order Information	
x_invoice_num	<p>Value: The merchant-assigned invoice number for the transaction.</p> <p>Format: 20-character maximum (no symbols).</p> <p>Notes: The invoice number must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>In order for this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>
x_description	<p>Value: The transaction description.</p> <p>Format: 255-character maximum (no symbols).</p> <p>Notes: The description must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>In order for this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>
Billing Information	
x_first_name	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The first name associated with the customer's billing address.</p> <p>Format: 50-character maximum (no symbols).</p>

Table 9 Payment Form Fields (Continued)

Field Name	Description
x_last_name	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The last name associated with the customer's billing address.</p> <p>Format: 50-character maximum (no symbols).</p>
x_company	<p>Value: The company associated with the customer's billing address.</p> <p>Format: 50-character maximum (no symbols).</p>
x_address	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The customer's billing address.</p> <p>Format: 60-character maximum (no symbols).</p> <p>Notes: Required if the merchant would like to use the Address Verification Service filter.</p> <p>For more information on AVS, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant</p> <p>Required for zero dollar authorizations for Visa verification transactions.</p>
x_city	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The city of the customer's billing address.</p> <p>Format: 40-character maximum (no symbols).</p>
x_state	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The state of the customer's billing address.</p> <p>Format: 40-character maximum (no symbols) or a valid 2-character state code</p>
x_zip	<p>Required when you use a European payment processor.</p> <p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The ZIP code of the customer's billing address.</p> <p>Format: 20-character maximum (no symbols).</p> <p>Notes: Required when the merchant uses the Address Verification Service filter.</p> <p>For more information on AVS, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant</p> <p>Required for zero dollar authorizations for Visa verification transactions.</p>

Table 9 Payment Form Fields (Continued)

Field Name	Description
x_country	<p>Required only when you use a European payment processor.</p> <p>Value: The country of the customer's billing address.</p> <p>Format: 60-character maximum (no symbols).</p>
x_phone	<p>Value: The phone number associated with the customer's billing address</p> <p>Format: 25-digit maximum (no letters).</p> <p>For example, (123)123-1234.</p>
x_fax	<p>Value: The fax number associated with the customer's billing address.</p> <p>Format: 25-digit maximum (no letters).</p> <p>For example, (123)123-1234.</p>
x_email	<p>Required only when you use a European payment processor.</p> <p>Value: The customer's valid email address.</p> <p>Format: 255-character maximum.</p> <p>For example, janedoe@customer.com.</p> <p>The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p> <p>For more information about Email Receipts, see the <i>Merchant Integration Guide</i>:</p> <p>http://www.authorize.net/support/merchant/</p>
x_cust_id	<p>Value: The merchant-assigned customer ID.</p> <p>Format: 20-character maximum (no symbols).</p> <p>Notes: The unique identifier to represent the customer associated with the transaction.</p> <p>The customer ID must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>In order for this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>
Shipping Information	
x_ship_to_first_name	<p>Value: The first name associated with the customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 50-character maximum (no symbols).</p>
x_ship_to_last_name	<p>Value: The last name associated with the customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 50-character maximum (no symbols).</p>

Table 9 Payment Form Fields (Continued)

Field Name	Description
x_ship_to_company	<p>Value: The company associated with the customer's shipping address.</p> <p>Format: 50-character maximum (no symbols).</p>
x_ship_to_address	<p>Value: The customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 60-character maximum (no symbols).</p>
x_ship_to_city	<p>Value: The city of the customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 40-character maximum (no symbols).</p>
x_ship_to_state	<p>Value: The state of the customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 40-character maximum (no symbols) or a valid two-character state code.</p>
x_ship_to_zip	<p>Value: The ZIP code of the customer's shipping address. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Format: 20-character maximum (no symbols).</p>
x_ship_to_country	<p>Value: The country of the customer's shipping address.</p> <p>Format: 60-character maximum (no symbols).</p>
Additional Shipping Information (Level 2 Data)	
x_tax	<p>Value: The valid tax amount OR delimited tax information.</p> <p>Format: When you submit delimited tax information, the field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total tax amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited tax information. This information includes:</p> <ul style="list-style-type: none"> ■ tax item name< > ■ tax description< > ■ tax amount <p>Format: The dollar sign (\$) is not allowed when you submit delimited information.</p> <p>Note: The total amount of the transaction in x_amount must include this amount.</p> <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_tax" VALUE="Tax1< >state tax< >0.0625"></pre>

Table 9 Payment Form Fields (Continued)

Field Name	Description
x_freight	<p>Value: The valid freight amount OR delimited freight information.</p> <p>Format: When you submit delimited freight information, field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total freight amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited freight information. This information includes:</p> <p>Delimited freight information fields include:</p> <ul style="list-style-type: none"> ■ freight item name< > Value: The freight item name. ■ freight item name< > Value: The freight item description. ■ freight amount Value: The freight amount. The total amount of the transaction in x_amount must <i>include</i> this amount. Format: The dollar sign (\$) is not allowed when submitting delimited information. <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_freight" VALUE="Freight1< >ground overnight< >12.95></pre>
x_duty	<p>Value: The valid duty amount OR delimited duty information.</p> <p>Format: When you submit delimited duty information, field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total duty amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited duty information. This information includes:</p> <ul style="list-style-type: none"> ■ duty item name< > Value: The duty item name. ■ duty description< > Value: The duty item description. ■ duty amount Value: The duty amount. The total amount of the transaction in x_amount must <i>include</i> this amount. Format: The dollar sign (\$) is not allowed when you submit delimited information. <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_duty" VALUE="Duty1< >export< > 15.00></pre>
x_tax_exempt	<p>Value: The tax exempt status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether the transaction is tax exempt.</p>

Table 9 Payment Form Fields (Continued)

Field Name	Description
x_po_num	<p>Value: The merchant-assigned purchase order number.</p> <p>Format: 25-character maximum (no symbols).</p> <p>Notes: The purchase order number must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>In order for this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p> <p>If your payment processor is EVO and you submit Level 2 data, you must also submit the x_po_num field.</p>

EVO Billing and Shipping Fields

If your payment processor is EVO and you submit any of the following billing fields, you must submit all of them. You must also set them as required fields in the Merchant Interface's Payment Form Settings page.

- x_first_name
- x_last_name
- x_address
- x_city
- x_state
- x_zip

If your payment processor is EVO and you submit one of the following shipping fields, you must submit all of them. You must also set them as required fields in the Merchant Interface's Payment Form Settings page.

- x_ship_to_first_name
- x_ship_to_last_name
- x_ship_to_address
- x_ship_to_city
- x_ship_to_state
- x_ship_to_zip

**Important**

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

**Note**

Delimited duty, freight, and tax information is 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.

Example 5 shows a request for additional supported fields using the hosted payment form. In this example, the payment form will display the Invoice Number, Description, Customer ID, billing information, and shipping information fields. You can also configure these fields for the payment form in the Merchant Interface.

**Note**

The Invoice Number and Customer ID must be created dynamically or provided per transaction in order for this information to be included in the post. The payment gateway does not perform this function.

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

**Note**

The code included in this document uses simulated field values in an ASP scripting environment. Code varies based on web programming language, so we do not recommend that you copy and paste it but rather use it as a guide.

Example 5 Payment Form Request with Additional Transaction Data

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

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

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

Configuring the Appearance of the Hosted Payment Form

You can configure the following hosted payment form settings to match the look of the merchant's web site:

- Text color
- Link text color
- Background color
- Header text (can include HTML)
- Footer text (can include HTML)
- Adding a Cancel link

You can configure the color and font settings in the Merchant Interface.

To configure color and font settings in the Merchant Interface:

- Step 1** Log in to the Merchant Interface.
 - Step 2** In the left menu, choose **Account > Settings**.
 - Step 3** In the Transaction Submission section, click **Receipt Page**.
 - Step 4** Click **Color and Font Settings** to open the color and font configuration page.
 - Step 5** Click **Help** to see complete instructions on how to use this page.
-

[Table 10](#) describes the fields that you can submit using the HTML Form POST to customize the merchant's payment form to look like your web site.

The form fields are submitted using the syntax:

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



Note

Fields that are submitted to the payment gateway using a transaction request will override field settings that are established in the Merchant Interface.

Table 10 Customizing the Hosted Payment Form

Field Name	Description
x_return_policy_url	The URL for the web page that describes the merchant's return policy.
x_header_html_payment_form	<p>Value: The hosted payment form header.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text submitted in this field is displayed as the header on the hosted payment form.</p>
x_footer_html_payment_form	<p>Value: The hosted payment form footer.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text submitted in this field is displayed as the footer on the hosted payment form.</p>
x_header2_html_payment_form	<p>Notes: Same as x_header_html_payment_form except that it appears at the very top of the page, above the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.</p>
x_footer2_html_payment_form	<p>Notes: Same as x_footer_html_payment_form, except that it appears at the very top of the page, above the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.</p>
x_color_background	<p>Value: The background color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: The value in this field sets the background color for the hosted payment form and receipt page.</p>
x_color_link	<p>Value: The hyperlink color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: The value of this field sets the color of the HTML links for the hosted payment form and the receipt page.</p>
x_color_text	<p>Value: The text color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: The value in this field sets the color of the text on the hosted payment form and the receipt page.</p>

Table 10 Customizing the Hosted Payment Form (Continued)

Field Name	Description
x_logo_url	<p>Value: The URL of the merchant's logo.</p> <p>Notes: The image referenced by this URL is displayed in the header of the hosted payment form and the receipt page.</p> <p>Logo images must be hosted on a secure server. See "Logos and Background Images for the Hosted Payment Form," page 49.</p>
x_background_url	<p>Value: The URL of the merchant's background image.</p> <p>Notes: The image referenced by this URL is displayed as the background on the hosted payment form and the receipt page.</p> <p>Background images must be hosted on a secure server. See "Logos and Background Images for the Hosted Payment Form," page 49.</p>
x_cancel_url	<p>Value: The URL to which the payment gateway redirects when the user clicks the Cancel link.</p> <p>Notes: An API parameter only and not available as a setting in the Merchant Interface.</p>
x_cancel_url_text	<p>Value: Custom text for the Cancel link.</p> <p>Format: The default value is Cancel.</p> <p>Notes: An API parameter only and not available as a setting in the Merchant Interface.</p>
x_font_family	<p>Value: Default font family for the Hosted Payment Form.</p> <p>Format: Follows the CSS 'font-family' standard.</p> <p>Notes: See http://www.w3.org/TR/CSS21/fonts.html#font-family-prop for details.</p> <p>Example: x_font_family="%22Courier%20New%22%2C%20Courier%2C%20monospace"</p>
x_font_size	<p>Value: Default font size for the Hosted Payment Form.</p> <p>Format: Expressed in points, suffixed with "px."</p> <p>Example: x_font_size="12px"</p>
x_sectionhead1_color_text	<p>Value: The text color for the header of the Hosted Payment Form.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Example: x_sectionhead1_color_text="%230000FF"</p>
x_sectionhead1_font_family	<p>Value: Font family for the header of the Hosted Payment Form.</p> <p>Format: Follows the CSS 'font-family' standard.</p> <p>Notes: See http://www.w3.org/TR/CSS21/fonts.html#font-family-prop for details.</p> <p>Example: x_sectionhead1_font_family="%22Courier%20New%22%2C%20Courier%2C%20monospace"</p>

Table 10 Customizing the Hosted Payment Form (Continued)

Field Name	Description
x_sectionhead1_font_size	<p>Value: Font size for the header of the Hosted Payment Form.</p> <p>Format: Expressed in points, suffixed with “px.”</p> <p>Example: x_sectionhead1_font_size="12px"</p>
x_sectionhead1_font_bold	<p>Value: Toggle for bold text for the header of the Hosted Payment Form.</p> <p>Format: TRUE, FALSE</p> <p>Example: x_sectionhead1_font_bold="TRUE"</p>
x_sectionhead1_font_italic	<p>Value: Toggle for italicized text for the header of the Hosted Payment Form.</p> <p>Format: TRUE, FALSE</p> <p>Example: x_sectionhead1_font_italic="FALSE"</p>



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

Placement of Custom Headers and Footers

The following images show where custom headers and footers appear on the hosted payment form.

Figure 3 Location of Custom Headers on Payment Form

The image shows a payment form with several sections and custom headers. The form is enclosed in a light gray border. At the top, there is a pink header bar with the text "x_header2_html_payment_form". Below this, the form is divided into sections by horizontal lines. The first section is "Order Information" with a "* Required Fields" note on the right. Below it is a "Description:" field. The second section is "Payment Information" which includes radio buttons for "Credit Card" (selected) and "Bank Account (USA only)". Below these are logos for "VISA" and "MasterCard". There are input fields for "Card Number:" (with a note "* (enter number without spaces or dashes)") and "Expiration Date:" (with a note "* (mmyy)" and the value "1220"). The third section is "Billing Information" which includes a "Customer ID: Customer" label and input fields for "First Name: First" and "Last Name: Last". A "Total: US \$4.50" label is positioned to the right of the "Description:" field.

Figure 4 Location of Custom Footers on Payment Form

The image shows a payment form with the following fields and sections:

- Phone: 2061111111
- Fax: 2062222222
- Shipping Information**
- Copy Billing Information to Shipping Information
- First Name: First
- Last Name: Last
- Company: Company
- Address: 123 Main St
- City: Test
- State/Province: WA
- Zip/Postal Code: 98004
- Country: USA

Below the shipping information section, there are two red horizontal bars indicating the location of custom footers:

- The first bar is labeled `x_footer_html_payment_form`.
- The second bar is labeled `x_footer2_html_payment_form`.

At the bottom of the form, there is a **Submit** button and a blue link labeled `x_cancel_url_text`.

Adding a Cancel Link

You can add a Cancel link to the hosted payment form that cancels the order. To do so, specify a value for the `x_cancel_url` field, which contains the URL to which the payment form returns customers when they click Cancel. See an example in Figure 4. You can also specify a value for `x_cancel_url_text`, which contains the text displayed on the Cancel link. The default text is Cancel.

Logos and Background Images for the Hosted Payment Form

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

To submit an image for hosting through the Merchant Interface:

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

Your request will be sent to our Customer Support department. Allow two business days for uploads to become available. When we have hosted your image, we will send you specific instructions on how to reference the file. To check the status of your eTicket at any time, log in to the Merchant Interface, click **Contact Us**, then click **Manage Existing eTickets**.

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

To submit an image for hosting through email:

- Step 1** Send an email with your request, your payment gateway ID, and the image file as an attachment to: support@authorize.net.

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

Image Requirements and Guidelines

Images must be in JPEG, GIF, or PNG formats. Other file formats will not be accepted.

Name the file using the convention *logo_GatewayID.ext*, where *GatewayID* is your payment gateway ID (6-digit maximum), and where *ext* is either *jpg*, *gif*, or *png*. For more information, see the [knowledge base article](#), "What is my Payment Gateway ID?"

If you have already submitted an image but have not received an update within two business days, contact Customer Support so we can verify that we received the image and have submitted it for hosting. We strongly recommend smaller files to ensure that your customers can view the full payment form quickly.

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

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

Merchant-Defined Fields

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

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



Merchant-Defined Data fields are not intended to and **MUST NOT** be used to capture personally identifying information. Accordingly, the merchant is prohibited from capturing, obtaining, and/or transmitting any personally identifying information in or by means of the Merchant-Defined Data fields. Personally identifying information includes, but is not limited to, name, address, credit card number, social security number, driver's license number, state-issued identification number, passport number, and card verification numbers (CVV, CVC2, CVV2, CID, CVN). If Authorize.Net discovers that the merchant is capturing and/or transmitting personally identifying information by means of the Merchant-Defined Data fields, whether or not intentionally, Authorize.Net **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.

Data submitted using merchant-defined fields is included in merchant confirmation emails (see "[Email Receipt](#)," page 62, for more information).



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

Receipt Options

In addition to the secure payment form, SIM provides two options for communicating the transaction results to the customer:

- 1 The payment gateway-hosted receipt page
- 2 Relay Response

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

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



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

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

Using the Hosted Receipt Page

You can configure settings for the hosted receipt page by passing fields in the transaction request per transaction, or in the Merchant Interface.



You should consider configuring these and other important integration settings using the HTML Form POST request. Doing so prevents the integration from being affected if these settings are inadvertently changed by the merchant in the Merchant Interface.

Receipt Link URL(s)

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

Receipt Method

This setting specifies the kind of link that directs the customer's browser to the merchant's web site.

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

For more information on configuring these settings in the Merchant Interface, see the *Merchant Integration Guide*:

<http://www.authorize.net/support/merchant/>.

Table 11 describes the form fields that you can submit in order to customize the hosted receipt page. Submit the form fields using this syntax:

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

Table 11 Customizing the Hosted Receipt Page

Field Name	Description
x_receipt_link_method	<p>Value: The type of link back to the merchant's web site from the hosted receipt page.</p> <p>Format: LINK, POST, or GET.</p> <p>Notes: LINK creates a hyperlink.</p> <p>GET creates a button and returns transaction information in the receipt link URL.</p> <p>POST creates a button and returns transaction information as an HTML Form POST.</p>

Table 11 Customizing the Hosted Receipt Page (Continued)

Field Name	Description
x_receipt_link_text	<p>Value: The text of the link or button that directs the customer back to the merchant's web site.</p> <p>Format: 50-character maximum.</p> <p>Notes: If the receipt link method is LINK, the field value is a hyperlinked text on the hosted receipt page. If the receipt link method is GET or POST, the field value becomes the text of a Submit button. An HTML form is created in the receipt page that has hidden fields containing the results of the transaction processed.</p>
x_receipt_link_url	<p>Value: The URL of the link or button that directs the customer back to the merchant's web site.</p> <p>Notes: For this field to be accepted as valid by the payment gateway, you must also configure the receipt link URL in the Merchant Interface. If the receipt link method is LINK, the URL specified becomes the href value of the hyperlinked text. If the receipt link method is GET or POST, the URL becomes the action of the HTML form.</p>
x_sectionhead2_color_text	<p>Value: The text color for the header of the Hosted Receipt Page.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Example: x_sectionhead2_color_text="%230000FF"</p>
x_sectionhead2_font_family	<p>Value: Font family for the header of the Hosted Receipt Page.</p> <p>Format: Follows the CSS 'font-family' standard.</p> <p>Notes: See http://www.w3.org/TR/CSS21/fonts.html#font-family-prop for details.</p> <p>Example: x_sectionhead2_font_family="%22Courier%20New%22%2C%20Courier%2C%20monospace"</p>
x_sectionhead2_font_size	<p>Value: Font size for the header of the Hosted Receipt Page.</p> <p>Format: Expressed in points, suffixed with "px."</p> <p>Example: x_sectionhead2_font_size="12px"</p>
x_sectionhead2_font_bold	<p>Value: Toggle for bold text for the header of the Hosted Receipt Page.</p> <p>Format: TRUE, FALSE</p> <p>Example: x_sectionhead2_font_bold="TRUE"</p>
x_sectionhead2_font_italic	<p>Value: Toggle for italicized text for the header of the Hosted Receipt Page.</p> <p>Format: TRUE, FALSE</p> <p>Example: x_sectionhead2_font_italic="FALSE"</p>

[Example 6](#) shows how to include a receipt link for the hosted receipt page in the transaction request.



Note

The code included in this document uses simulated field values in an ASP scripting environment. Code varies based on web programming language, so we do not recommend that you copy and paste it but rather use it as a guide.

Example 6 Payment Form Request Including Receipt Link URL

```

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

```

The example above produces a button that is displayed to the customer upon checkout (see [Figure 1, page 33](#)). When the customer clicks the button, the secure hosted payment form is displayed (see [Figure 2, page 34](#)).

After the customer submits the transaction, the hosted receipt page ([Figure 5](#)) appears.

Figure 5 The Hosted Receipt Page with Receipt Link URL

[Click here to return to our homepage.](#)

Thank you for your order!

You may print this receipt page for your records. A receipt has also been emailed to you.

Order Information

Merchant: Business Name
 Description:
 Date/Time: 28-Jun-2011 09:47:32 AM PT Invoice Number:
 Customer ID: Customer

Billing Information	Shipping Information
First Last Company 123 Main St Test, WA 98004 USA blackhole@authorize.net Phone: 2061111111 Fax: 2062222222	First Last Company 123 Main St Test, WA 98004 USA

Total: US \$4.50

Visa **0027**

Date/Time: 28-Jun-2011 09:47:32 AM PT
 Transaction ID: 2148221874
 Authorization Code: VYBR5B
 Payment Method: Visa ****0027

**Important**

Submitting these fields simply places the receipt link URL on the receipt page. To customize the placement of a URL on the receipt page, reference it in HTML in either the receipt page header or footer API fields (**x_header_html_receipt**, **x_footer_html_receipt**) or in the Merchant Interface receipt page header and footer settings.

Customizing the Receipt Page

When you use the hosted receipt page, you can configure the following settings to match the look of the merchant's web site.

- Text color
- Link text color
- Background color
- Header text (can include HTML)
- Footer text (can include HTML)

To configure these settings in the Merchant Interface:

- Step 1** Log in to the [Merchant Interface](#).
- Step 2** In the left menu, choose **Account > Settings**.
- Step 3** Choose one of these options:
- In the Transaction Format section, click **Payment Form**
 - In the Transaction Submission section, click **Receipt Page**
- Step 4** Click **Color and Font Settings** to open the color and font configuration page.
- Step 5** Click **Help** to see complete instructions on how to use this page.

[Table 12](#) describes the fields that you can submit by means of the HTML Form POST to customize the merchant's payment form to look like their web site.

Submit the form fields using the following syntax:

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



Note

Fields submitted to the payment gateway using a transaction request will override field settings configured in the Merchant Interface.

Table 12 Customizing the Receipt Page

Field Name	Description
x_header_html_receipt	<p>Value: The hosted receipt page header.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text that you submit in this field is displayed at the top of the hosted receipt page.</p>
x_footer_html_receipt	<p>Value: The hosted receipt page footer.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text that you submit in this field is displayed at the bottom of the hosted receipt page.</p>
x_header2_html_receipt	<p>Notes: Same as x_header_html_receipt except that it is displayed at the very top of the page, above the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.</p>
x_footer2_html_receipt	<p>Notes: Same as x_footer_html_receipt, except that it is displayed at the very bottom of the page, below the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.</p>
x_color_background	<p>Value: The background color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: This field is common to the hosted payment form and receipt page. The value in this field sets the background color for both.</p>

Table 12 Customizing the Receipt Page (Continued)

Field Name	Description
x_color_link	<p>Value: The hyperlink color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: This field is common to the hosted payment form and receipt page. The value in this field sets the color of the HTML links for both.</p>
x_color_text	<p>Value: The text color.</p> <p>Format: Any valid HTML color name or color hex code.</p> <p>Notes: This field is common to the hosted payment form and receipt page. The value in this field sets the color of the text on the hosted payment form and the receipt page.</p>
x_logo_url	<p>Value: The URL of the merchant's logo.</p> <p>Notes: The image referenced by this URL is displayed on the header of the hosted payment form and receipt page.</p> <p>Logo images must be uploaded to the payment gateway server. See "Logos and Background Images for the Hosted Payment Form," page 49.</p>
x_background_url	<p>Value: The URL of the merchant's background image.</p> <p>Notes: The image referenced by this URL is displayed as the background of the hosted payment form and receipt page.</p> <p>Background images must be uploaded to the payment gateway server. See "Logos and Background Images for the Hosted Payment Form," page 49.</p>

Logos and Background Images for the Hosted Receipt Page

If the merchant is using a logo and/or background images on the hosted payment form, the same image files can be referenced for display on the hosted receipt page. Image files must be uploaded to the payment gateway server in order to be displayed properly. For more information on how to upload image files, see ["Logos and Background Images for the Hosted Payment Form," page 49.](#)

Relay Response

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

[Table 13](#) describes form fields that you can submit in order to configure Relay Response. Except for **x_relay_always**, you can also configure these settings in the Merchant Interface. For more information about configuring Relay Response in the Merchant Interface, see the *Merchant Integration Guide*:

<http://www.authorize.net/support/merchant/>

Use this syntax to submit the form fields:

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

Table 13 Configuring Relay Response

Field Name	Description
x_relay_response	<p>Value: The request for a relay response.</p> <p>Format: TRUE, FALSE</p> <p>Notes: This field instructs the payment gateway to return transaction results to the merchant using an HTML form POST to the merchant's web server for a relay response.</p>
x_relay_always	<p>Value: Requests a relay response even for partial authorizations and in case of errors.</p> <p>Format: TRUE, FALSE</p> <p>Notes: This field instructs the payment gateway to return a relay response regardless of any declines, errors, or partial authorizations.</p>
x_relay_url	<p>Value: The URL on the merchant's web site to which the payment gateway posts transaction results for a relay response.</p> <p>Format: Any valid URL. Including name/value pairs in the URL (anything after a question mark (?)) is not recommended.</p> <p>Notes: If you submit this field, the payment gateway validates the URL value against the Relay Response URL configured in the Merchant Interface. If the URL submitted does not match the URL configured in the Merchant Interface, the transaction is rejected. If no value is submitted in the HTML Form POST, the payment gateway posts transaction results to the URL configured in the Merchant Interface.</p>



Note

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

Example 7 shows how to include the Relay Response request in the HTML Form POST.



Note

The code included in this document uses simulated field values in an ASP scripting environment. Code varies based on web programming language, so we do not recommend that you copy and paste it but rather use it as a guide.

Example 7 Payment Form Request Including Relay Response Request

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

```
<INPUT TYPE=HIDDEN NAME="x_login" VALUE="the merchant's API Login ID">
<INPUT TYPE=HIDDEN NAME="x_version" VALUE="3.1">
<INPUT TYPE=HIDDEN NAME="x_method" VALUE="CC">
<INPUT TYPE=HIDDEN NAME="x_show_form" VALUE="PAYMENT_FORM">
<INPUT TYPE=HIDDEN NAME="x_amount" VALUE="9.95">
<INPUT TYPE=HIDDEN NAME="x_relay_response" VALUE="TRUE">
<INPUT TYPE=HIDDEN NAME="x_relay_url" VALUE="Any valid URL">
<INPUT TYPE=SUBMIT VALUE="Click here for the secure payment form">
</FORM>
```

When Authorize.Net sends a Relay Response to the merchant's server, and the merchant's web server does not send a positive response within 10 seconds, the connection times out and an error is generated for the transaction.



Note

All web traffic to and from Authorize.Net must use ports 80 and 443.

Whitelisting

There are two ways that you can specify the relay response URL. First, you can send it in the **x_relay_url** field. Second, you can specify it on the Response/Receipt URLs page in the Merchant Interface as the default relay response URL. If you specify a default relay response URL, you don't have to specify **x_relay_url** in your code, and the response will always post to this default URL.

The list of URLs entered in the Response/Receipt URLs page in the Merchant Interface also acts as a whitelist of allowed relay or receipt URLs. If you submit a URL in the **x_relay_url** field that is not specified in the Response/Receipt URLs page in the Merchant Interface, an error is displayed.

A whilelist is a list of allowed values.

Tips for Using Relay Response

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

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

The relay response is rendered on the payment gateway server. Custom receipt pages **must** incorporate absolute URLs.

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

Email Receipt

Merchants can send an email receipt generated by the payment gateway to customers who provide an email address with their transaction. The email receipt includes a summary and results of the transaction. To the customer, this email appears to be sent from the merchant contact that is configured as the email sender in the Merchant Interface. For more information about the Email Sender setting, see the *Merchant Integration Guide*:

<http://www.authorize.net/support/merchant/>

To send the email receipt generated by the payment gateway, submit the following API fields with the transaction request string or configure them in the Merchant Interface.

Submit the API form fields using the following syntax:

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

Table 14 Configuring the Customer Email Receipt

Field Name	Description
x_email	<p>Value: The customer's valid email address.</p> <p>Format: 255-character maximum.</p> <p>For example: janedoe@customer.com.</p> <p>Notes: The email address to which the customer's copy of the email receipt is sent when the Email Receipts setting is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p>
x_email_customer	<p>Value: The customer email receipt status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether an email receipt should be sent to the customer.</p> <p>If set to TRUE, the payment gateway sends an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer.</p> <p>If no value is submitted, the payment gateway looks up the configuration in the Merchant Interface and sends an email only if the merchant has enabled the setting. If this field is not submitted, and the setting is disabled in the Merchant Interface, no email is sent.</p>
x_header_email_receipt	<p>Value: The email receipt header.</p> <p>Format: Plain text.</p> <p>Notes: This text appears as the header of the email receipt sent to the customer.</p>
x_footer_email_receipt	<p>Value: The email receipt footer.</p> <p>Format: Plain text.</p> <p>Notes: This text appears as the footer on the email receipt sent to the customer.</p>

In addition, the merchant can receive a transaction confirmation email from the payment gateway at the completion of each transaction, which includes order information and the results of the transaction. Merchants can enroll for confirmation emails in the Merchant Interface.

Additional API Fields

[Table 15](#) and [Table 16](#) describe API fields that can be submitted in a transaction request to the payment gateway in addition to the minimum required fields. Submit form fields using the syntax:

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

Transaction Information

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

Table 15 Fields Containing Transaction-Specific Information

Field Name	Description
x_version	<p>Value: The merchant's transaction version.</p> <p>Format: 3.0, 3.1</p> <p>Notes: Indicates to the system the set of fields to be included in the response; 3.0 is the default version.</p> <p>3.1 enables the merchant to use the Card Code feature, and it is the current standard version.</p> <p>We recommend that you submit this field per transaction, particularly if you are using Relay Response. For more information, see "SIM Relay Response," page 76, and Appendix A, "Fields by Transaction Type," on page 91.</p>
x_method	<p>Value: The payment method.</p> <p>Format: CC or ECHECK.</p> <p>Notes: The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If left blank, this value defaults to CC.</p> <p>For more information about eCheck.Net transaction requirements, see the <i>eCheck.Net Developer Guide</i>: http://developer.authorize.net/guides/echeck.pdf</p>

Table 15 Fields Containing Transaction-Specific Information (Continued)

Field Name	Description
x_test_request	<p>Value: The request to process test transactions.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p> <p>Notes: Indicates whether the transaction should be processed as a test transaction. See "Testing Your Integration," page 90, for more information.</p>
x_duplicate_window	<p>Value: The period of time after a transaction is submitted during which a duplicate transaction cannot be submitted.</p> <p>Format: Any value from 0 through 28800 (no comma).</p> <p>Notes: Indicates in seconds the period of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction. The maximum time allowed is 8 hours (28800 seconds).</p> <p>If a value less than 0 is sent, the payment gateway defaults to 0 seconds. If a value greater than 28800 is sent, the payment gateway defaults to 28800. If no value is sent, the payment gateway defaults to 2 minutes (120 seconds).</p> <p>If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See "Response for Duplicate Transactions," page 75, for more information.</p>

Itemized Order Information

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

The value for the **x_line_item** field can include delimited item information. Item information must be delimited by a bracketed pipe <|>. Line item values must be included in the order in which they are listed in [Table 16](#).

[Table 16](#) describes the item information elements of the **x_line_item** field. A code example is presented after the table.

Table 16 Delimited x_line_item Information

Item Information Elements	Description
item ID< >	<p>Format: 31-character maximum.</p> <p>Notes: ID assigned to an item.</p>
item name< >	<p>Format: 31-character maximum.</p> <p>Notes: Short description of an item.</p>

Table 16 Delimited x_line_item Information (Continued)

Item Information Elements	Description
item description< >	Format: 255-character maximum. Notes: Detailed description of an item.
item quantity< >	Format: Maximum of 2 decimal places. Must be a positive number. Notes: Quantity of an item.
item price (unit cost)< >	Format: Maximum of 2 decimal places. Must be a positive number. The dollar sign (\$) is not allowed when you submit delimited information. Notes: Cost of an item per unit, excluding tax, freight, and duty.
item taxable	Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0 Notes: Optional. False by default. Indicates whether the item is subject to tax.

The merchant can submit as many as 30 distinct line items containing itemized order information per transaction. All field separators are required whether the field has a value or not. In [Example 8](#), the item description field after *golf balls*<|> has no value, yet the bracketed pipe remains.

Example 8 Submitting Itemized Order Information

```
x_line_item=item1<|>golf balls<|><|>2<|>18.95<|>Y
```

```
x_line_item=item2<|>golf bag<|>Wilson golf carry bag,  
red<|>1<|>39.99<|>Y&
```

```
x_line_item=item3<|>book<|>Golf for Dummies<|>1<|>21.99<|>Y
```



Note

For Prior Authorization and Capture transactions, if line item information is submitted with the original transaction, you can submit adjusted information if the transaction changes. If you do not submit adjusted line item information, the information submitted with the original transaction applies.

Additional Customer Information

x_customer_ip

Value: The customer's IP address.

Format: 15-character maximum (no letters).

For example: 255.255.255.255.

Notes: IP address of the customer initiating the transaction. If this value is not passed, it defaults to 255.255.255.255.

This field is required only when you are using customer IP-based Advanced Fraud Detection Suite™ (AFDS) filters. For more information about AFDS, see the *Merchant Integration Guide*:

<http://www.authorize.net/support/merchant/>

Transaction Response

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

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

Fields in the Payment Gateway Response

Table 17 lists the fields returned in the response from the payment gateway.

Transaction response fields are not necessarily sent in the exact order listed here. Developers are encouraged to use the name of the field in order to find the correct response. If your code requires transaction response fields in a particular order, future updates to the SIM API may cause unexpected results from your code.

Table 17 Fields in the Payment Gateway Response

Field Name	Description
x_response_code	<p>Value: The overall status of the transaction.</p> <p>Format:</p> <ul style="list-style-type: none"> 1—Approved 2—Declined 3—Error 4—Held for Review
x_response_reason_code	<p>Value: A code that corresponds to more details about the result of the transaction.</p> <p>Format: Numeric.</p> <p>Notes: See "Response Code Details," page 77, for a listing of response reason codes.</p>

Table 17 Fields in the Payment Gateway Response (Continued)

Field Name	Description
x_response_reason_text	<p>Value: A brief description of the result, which corresponds with the response reason code.</p> <p>Format: Text.</p> <p>Notes: You can generally use this text to display a transaction result or error to the customer. However, review "Response Code Details," page 77, to identify any specific text that you do not want to pass to the customer.</p>
x_auth_code	<p>Value: The authorization or approval code.</p> <p>Format: 6 characters.</p>
x_avs_code	<p>Value: The Address Verification Service (AVS) response code.</p> <p>Format:</p> <p>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 unavailable. W— 9-digit ZIP matches, Address (Street) does not. X—Address (Street) and nine digit ZIP match. Y—Address (Street) and five digit ZIP match. Z—5-digit ZIP matches, Address (Street) does not.</p> <p>Notes: Indicates the result of the (AVS) filter.</p> <p>For more information about AVS, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>
x_trans_id	<p>Value: The payment-gateway-assigned identification number for the transaction.</p> <p>Format: When x_test_request is submitted, this value will be 0.</p>
x_invoice_num	<p>Value: The merchant-assigned invoice number for the transaction.</p> <p>Format: 20-character maximum (no symbols).</p>
x_description	<p>Value: The transaction description.</p> <p>Format: 255-character maximum (no symbols).</p>
x_amount	<p>Value: The amount of the transaction.</p> <p>Format: 15-digit maximum.</p>
x_method	<p>Value: The payment method.</p> <p>Format: CC or ECHECK</p>

Table 17 Fields in the Payment Gateway Response (Continued)

Field Name	Description
x_type	Value: The type of credit card transaction. Format: AUTH_CAPTURE, AUTH_ONLY
x_account_number	Value: Last 4 digits of the card provided. Format: Alphanumeric (XXXX6835)
x_card_type	Value: Visa, MasterCard, American Express, Discover, Diners Club, JCB. Format: Text
x_split_tender_id	Value: Value that links the current authorization request to the original authorization request. This value is returned in the reply message from the original authorization request. Format: Alphanumeric. Notes: Returned in the reply message for the first transaction that receives a partial authorization.
x_prepaid_requested_amount	Value: Amount requested in the original authorization. Format: Numeric. Notes: Present if the current transaction is for a prepaid card or if a split-tender ID was sent in.
x_prepaid_balance_on_card	Value: Balance on the debit card or prepaid card. Format: Numeric Notes: Present if the current transaction is for a prepaid card or if a split-tender ID was sent in.
x_cust_id	Value: The merchant-assigned customer ID. Format: 20-character maximum (no symbols).
x_first_name	Value: The first name associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_last_name	Value: The last name associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_company	Value: The company associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_address	Value: The customer's billing address. Format: 60-character maximum (no symbols).
x_city	Value: The city of the customer's billing address. Format: 40-character maximum (no symbols).
x_state	Value: The state of the customer's billing address. Format: 40-character maximum (no symbols) or a valid 2-character state code
x_zip	Value: The ZIP code of the customer's billing address. Format: 20-character maximum (no symbols).

Table 17 Fields in the Payment Gateway Response (Continued)

Field Name	Description
x_country	Value: The country of the customer's billing address. Format: 60-character maximum (no symbols).
x_phone	Value: The phone number associated with the customer's billing address. Format: 25-digit maximum (no letters). For example, (123)123-1234.
x_fax	Value: The fax number associated with the customer's billing address. Format: 25-digit maximum (no letters). For example, (123)123-1234.
x_email	Value: The customer's valid email address. Format: 255-character maximum.
x_ship_to_first_name	Value: The first name associated with the customer's shipping address. Format: 50-character maximum (no symbols).
x_ship_to_last_name	Value: The last name associated with the customer's shipping address. Format: 50-character maximum (no symbols).
x_ship_to_company	Value: The company associated with the customer's shipping address. Format: 50-character maximum (no symbols).
x_ship_to_address	Value: The customer's shipping address. Format: 60-character maximum (no symbols).
x_ship_to_city	Value: The city of the customer's shipping address. Format: 40-character maximum (no symbols).
x_ship_to_state	Value: The state of the customer's shipping address. Format: 40-character maximum (no symbols) or a valid 2-character state code.
x_ship_to_zip	Value: The ZIP code of the customer's shipping address. Format: 20-character maximum (no symbols).
x_ship_to_country	Value: The country of the customer's shipping address. Format: 60-character maximum (no symbols).
x_tax	Value: The tax amount charged. Format: Numeric. Notes: Delimited tax information is not included in the transaction response.
x_duty	Value: The duty amount charged. Format: Numeric. Notes: Delimited duty information is not included in the transaction response.
x_freight	Value: The freight amount charged. Format: Numeric. Notes: Delimited freight information is not included in the transaction response.

Table 17 Fields in the Payment Gateway Response (Continued)

Field Name	Description
x_tax_exempt	<p>Value: The tax exempt status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0</p>
x_po_num	<p>Value: The merchant-assigned purchase order number.</p> <p>Format: 25-character maximum (no symbols).</p>
x_MD5_Hash	<p>Value: The payment gateway generated MD5 hash value that can be used to authenticate the transaction response.</p> <p>Notes: For more information about creating an MD5 hash value, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>
x_cvv2_resp_code	<p>Value: The card code verification (CCV) response code</p> <p>Format:</p> <p>M—Match N—No Match P—Not Processed S—Should have been present U—Issuer unable to process request</p> <p>Notes: Indicates the result of the CCV filter.</p> <p>For more information about CCV, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>
x_cavv_response	<p>Value: The cardholder authentication verification response code.</p> <p>Format: Blank or not present—CAVV not validated.</p> <p>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.</p> <p>Notes: The cardholder authentication programs do not apply to SIM.</p>

Using the HMAC SHA-512 Hash Feature



Note

Using MD5 instead of HMAC512 is still supported but not recommended.

You can use an HMAC-SHA512 hash to authenticate responses. Use the signature key as the key in an HMAC computation (FIPS PUB 198-1) with SHA-512 (FIPS PUB 180-4). The response fields will be incorporated into the calculation in the sequence shown below, with each field encapsulated by a caret (^) character. If a field is empty, it must be qualified with a caret separator. A hash will always incorporate at least 31 caret (^) characters, even if only 5 fields are provided.



Important

The signature key is represented in the hexadecimal format. You must convert it to bytes before using it as key for hash calculation. Submitting the signature key in hexadecimal format results in an error.

Table 18 Order of API Fields in the Response

Order Number	Response Field Name	Order Number	Response Field Name
1	x_trans_id	16	x_state
2	x_test_request	17	x_zip
3	x_response_code	18	x_country
4	x_auth_code	19	x_phone
5	x_cvv2_resp_code	20	x_fax
6	x_cavv_response	21	x_email
7	x_avs_code	22	x_ship_to_company
8	x_method	23	x_ship_to_first_name
9	x_account_number	24	x_ship_to_last_name
10	x_amount	25	x_ship_to_address
11	x_company	26	x_ship_to_city
12	x_first_name	27	x_ship_to_state
13	x_last_name	28	x_ship_to_zip
14	x_address	26	x_ship_to_country
15	x_city	30	x_invoice_num

Example Input String

```
^103^false^1^F3K0UI^P^2^Y^CC^XXXX1111^1.02^Good
Company^FirstName^LastName^123 Elm
Street^Seattle^WA^98112^USA^4251234567^^^^^^^^^^946155502664743^
```

Example Sample Hash

```
4B9681F27142CC3A93B608B89CAA26A15F7ADDC36776D87D87C9B41F176EFFE536
F902EA36469FCC7162FE9872CA2EC6F236FCFD200B890B7B4085B79EF5EC3D
```

Example Hash calculation

```

        public string HMACSHA512(string key, string textToHash)
    {
        if (string.IsNullOrEmpty(key))
            throw new ArgumentNullException("HMACSHA512: key", "Parameter cannot be empty.");

        if (string.IsNullOrEmpty(textToHash))
            throw new ArgumentNullException("HMACSHA512: textToHash", "Parameter cannot be empty.");

        if (key.Length % 2 != 0 || key.Trim().Length < 2)
        {
            throw new ArgumentNullException("HMACSHA512: key", "Parameter cannot be odd or less than 2 characters.");
        }

        try
        {
            byte[] k = Enumerable.Range(0, key.Length)
                .Where(x => x % 2 == 0)
                .Select(x => Convert.ToByte(key.Substring(x, 2),
16))
                .ToArray();

            //Please note: the following code will not work.
            //byte[] k = (new System.Text.ASCIIEncoding()).GetBytes(key);

            HMACSHA512 hmac = new HMACSHA512(k);
            byte[] HashedValue = hmac.ComputeHash((new System.Text.ASCIIEncoding()).GetBytes(textToHash));

            return BitConverter.ToString(HashedValue).Replace("-", string.Empty);
        }
        catch (Exception ex)
    }

```

```
    {  
        throw new Exception("HMACSHA512: " + ex.Message);  
    }  
}
```

Response for Duplicate Transactions

The SIM API enables you to specify the period after a transaction is submitted during which the payment gateway checks for a duplicate transaction (based on credit card number, invoice number, amount, billing address information, transaction type, etc.) using the duplicate window field (**x_duplicate_window**). The value for this field can be from 0 through 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 period 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 period 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 is declined, and a value is 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
- MD5 hash (if this feature was used for the original transaction)

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

SIM Relay Response

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

SIM Transaction Response Versions

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

Version 3.0

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

Version 3.1

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

Upgrading the Transaction Version

To upgrade the transaction version, follow these steps (only users with the appropriate permissions can access this setting).

To upgrade the transaction version:

- Step 1** Log on to the Merchant Interface.
- Step 2** From the main menu, choose **Settings**.
- Step 3** In the Transaction Response section, click **Transaction Version**.
- Step 4** Change the transaction version using the drop-down box.
- Step 5** Click **Submit**.



Note

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

Response Code Details

The following tables describe the response codes and response reason texts that are returned for each transaction. The Authorize.Net Developer Center provides a valuable tool for troubleshooting:

<http://developer.authorize.net/tools/responsereasoncode>

Response Codes

A response code indicates the overall status of the transaction with possible values of approved, declined, error, or held for review.

Table 19 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

A 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 or customer or both to provide more information about the status of the transaction.

Table 20 Response Reason Codes and Response Reason Text

Response Code	Response Reason Code	Response Reason Text	Notes
1	1	This transaction has been approved.	
2	2	This transaction has been declined.	
2	3	This transaction has been declined.	
2	4	This transaction has been declined.	The code returned from the processor indicating that the card used needs to be picked up.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	5	A valid amount is required.	The value submitted in the amount field did not pass validation for a number.
3	6	The credit card number is invalid.	
3	7	The credit card expiration date is invalid.	The format of the date submitted was incorrect.
3	8	The credit card has expired.	
3	9	The ABA code is invalid.	The value submitted in the x_bank_aba_code field did not pass validation or was not for a valid financial institution.
3	10	The account number is invalid.	The value submitted in the x_bank_acct_num field did not pass validation.
3	11	A duplicate transaction has been submitted.	A transaction with identical amount payment information was submitted during the duplicate transaction period for the original transaction. See "Response for Duplicate Transactions," page 75 , for more details.
3	12	An authorization code is required but not present.	A transaction that required x_auth_code to be present was submitted without a value.
3	13	The merchant API Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applies 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 (such as VOID, PRIOR_AUTH_CAPTURE, and CREDIT).
3	16	The transaction was not found.	The transaction ID sent in was properly formatted but the gateway had no record of the transaction for the gateway account used.
3	17	The merchant does not accept this type of credit card.	The merchant was not configured to accept the credit card type submitted in the transaction.
3	18	ACH transactions are not accepted by this merchant.	The merchant does not accept electronic checks.
3	19 - 23	An error occurred during processing. Please try again in 5 minutes.	

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	24	The Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
3	25 - 26	An error occurred during processing. Please try again in 5 minutes.	
2	27	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	
2	28	The merchant does not accept this type of credit card.	The merchant ID at the processor was not configured to accept this card type.
2	29	The Paymentech identification numbers are incorrect. Call Merchant Service Provider.	
2	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
2	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly configured 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 " Form Settings " in the Merchant Integration Guide for details.
2	34	The VITAL identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly configured at the processor.
2	35	An error occurred during processing. Call Merchant Service Provider.	The merchant was incorrectly configured 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 configured at the processor.
3	40	This transaction must be encrypted.	
2	41	This transaction has been declined.	This code is returned if a transaction's fraud score is higher than the threshold set by the merchant.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	43	The merchant was incorrectly set up at the processor. Call your Merchant Service Provider.	The merchant was incorrectly configured at the processor.
2	44	This transaction has been declined.	The card code submitted with the transaction did not match the card code on file at the card issuing bank, and the transaction was declined.
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	The merchant tried 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 be performed only against settled transactions. The transaction against which the credit/refund was submitted has not been settled, so a credit cannot be issued.
3	51	The sum of all credits against this transaction is greater than the original transaction amount.	
3	52	The transaction was authorized, but the client could not be notified; the transaction will not be settled.	
3	53	The transaction type was invalid for ACH transactions.	If x_method = ECHECK, x_type cannot be set to CAPTURE_ONLY.
3	54	The referenced transaction does not meet the criteria for issuing a credit.	
3	55	The sum of credits against the referenced transaction would exceed the original debit amount.	The transaction is rejected if the sum of this credit and prior credits exceeds the original debit amount.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	56	This merchant accepts ACH transactions only; no credit card transactions are accepted.	The merchant processes eCheck.Net transactions only and does not accept credit cards.
3	57 - 63	An error occurred in processing. Please try again in 5 minutes.	
2	65	This transaction has been declined.	The transaction was declined because the merchant configured their account through the Merchant Interface to reject transactions with certain values for a Card Code mismatch.
3	66	This transaction cannot be accepted for processing.	The transaction did not meet gateway security guidelines.
3	68	The version parameter is invalid.	The value submitted in x_version was invalid.
3	69	The transaction type is invalid.	The value submitted in x_type was invalid.
3	70	The transaction method is invalid.	The value submitted in x_method was invalid.
3	71	The bank account type is invalid.	The value submitted in x_bank_acct_type was invalid.
3	72	The authorization code is invalid.	The value submitted in x_auth_code was more than 6 characters long.
3	73	The driver's license date of birth is invalid.	The format of the value submitted in x_drivers_license_dob was invalid.
3	74	The duty amount is invalid.	The value submitted in x_duty failed format validation.
3	75	The freight amount is invalid.	The value submitted in x_freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in x_tax failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in x_customer_tax_id failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in x_card_code failed format validation.
3	79	The driver's license number is invalid.	The value submitted in x_drivers_license_num failed format validation.
3	80	The driver's license state is invalid.	The value submitted in x_drivers_license_state failed format validation.
3	81	The requested form type is invalid.	The merchant requested an integration method not compatible with the AIM API.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	82	Scripts are only supported in version 2.5.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	83	The requested script is either invalid or no longer supported.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	84	This reason code is reserved or not applicable to this API.	
3	85	This reason code is reserved or not applicable to this API.	
3	86	This reason code is reserved or not applicable to this API.	
3	87	This reason code is reserved or not applicable to this API.	
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.	Applies only to SIM API. Fingerprints are valid only for a short period of time. This code indicates that the transaction fingerprint has expired.
3	98	This transaction cannot be accepted.	Applies only to SIM API. The transaction fingerprint has already been used.
3	99	This transaction cannot be accepted.	Applies only to SIM API. The server-generated fingerprint does not match the merchant-specified fingerprint in the x_fp_hash field.
3	100	The eCheck.Net type is invalid.	Applies only to eCheck.Net. The value specified in the x_echeck_type field is invalid.
3	101	The given name on the account and/or the account type does not match the actual account.	Applies only to eCheck.Net. The specified name on the account or the account type or both do not match the NOC record for this account.
3	102	This request cannot be accepted.	A password or Transaction Key was submitted with this webLink request. This is a high security risk.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.	Applies only to eCheck.Net. The value submitted for country failed validation.
3	105	This transaction is currently under review.	Applies only to eCheck.Net. The values submitted for city and country failed validation.
3	106	This transaction is currently under review.	Applies only to eCheck.Net. The value submitted for company failed validation.
3	107	This transaction is currently under review.	Applies only to eCheck.Net. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applies only to eCheck.Net. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applies only to eCheck.Net. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applies only to eCheck.Net. The value submitted for bank account name does not contain valid characters.
3	120	An error occurred during processing. Please try again.	The system-generated void for the original timed-out transaction failed. (The original transaction timed out while waiting for a response from the authorizer.)
3	121	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a database error.)
3	122	An error occurred during processing. Please try again.	The system-generated void for the original errored transaction failed. (The original transaction experienced a processing error.)
3	123	This account has not been given the permission(s) required for this request.	The transaction request must include the API login ID associated with the payment gateway account.
2	127	The transaction resulted in an AVS mismatch. The address provided does not match billing address of cardholder.	The system-generated void for the original AVS-rejected transaction failed.
3	128	This transaction cannot be processed.	The customer's financial institution does not currently allow transactions for this account.
3	130	This payment gateway account has been closed.	IFT: the payment gateway account status is Blacklisted.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	131	This transaction cannot be accepted at this time.	IFT: the payment gateway account status is Suspended-STA.
3	132	This transaction cannot be accepted at this time.	IFT: the payment gateway account status is Suspended-Blacklist.
2	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 of credit transactions. Debit the credit card instead of voiding the credit.
3	180	An error occurred during processing. Please try again.	The processor response format is invalid.
3	181	An error occurred during processing. Please try again.	The system-generated void for the original invalid transaction failed. The original transaction included an invalid processor response format.
3	185	This reason code is reserved or not applicable to this API.	
4	193	The transaction is currently under review.	The transaction was placed under review by the risk management system.
2	200	This transaction has been declined.	This error code applies only to merchants on FDC Omaha. The credit card number is invalid.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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.
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.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
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 x_recurring_billing and x_echeck_type is not allowed.
3	244	This eCheck.Net type is not allowed for this Bank Account Type.	The combination of values submitted for x_bank_acct_type and x_echeck_type is not allowed.
3	245	This eCheck.Net type is not allowed when using the payment gateway hosted payment form.	The value submitted for x_echeck_type is not allowed with the payment gateway hosted payment form.
3	246	This eCheck.Net type is not allowed.	The merchant's payment gateway account is not enabled to submit the eCheck.Net type.
3	247	This eCheck.Net type is not allowed.	The combination of values submitted for x_type and x_echeck_type is not allowed.

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
2	250	This transaction has been declined.	This transaction was submitted from a blocked IP address.
2	251	This transaction has been declined.	The transaction was declined because a Fraud Detection Suite filter was triggered.
4	252	Your order has been received. Thank you for your business!	The transaction was accepted but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
4	253	Your order has been received. Thank you for your business!	The transaction was accepted and was authorized but is being held for merchant review. The merchant can customize the customer response in the Merchant Interface.
2	254	Your transaction has been declined.	The transaction was declined after manual review.
3	261	An error occurred during processing. Please try again.	The transaction experienced an error during sensitive data encryption and was not processed. Try again.
3	270	The line item [item number] is invalid.	A value submitted in x_line_item for the item referenced is invalid.
3	271	The number of line items submitted is not allowed. A maximum of 30 line items can be submitted.	The number of line items submitted exceeds the allowed maximum of 30.
3	288	Merchant is not registered as a Cardholder Authentication participant. This transaction cannot be accepted.	The merchant has not indicated participation in any Cardholder Authentication Programs in the Merchant Interface.
3	289	This processor does not accept zero dollar authorization for this card type.	Your credit card processing service does not yet accept zero dollar authorizations for Visa credit cards. You can find your credit card processor listed on your merchant profile.
3	290	One or more required AVS values for zero dollar authorization were not submitted.	When submitting authorization requests for Visa, the address and zip code fields must be entered.
4	295	The amount of this request was only partially approved on the given prepaid credit card. A second credit card is required to complete the balance of this transaction.	The amount authorized is less than the requested transaction amount.
3	296	The specified SplitTenderId is not valid.	

Table 20 Response Reason Codes and Response Reason Text (Continued)

Response Code	Response Reason Code	Response Reason Text	Notes
3	297	A Transaction ID and a Split Tender ID cannot both be used in a single transaction request.	
3	300	The device ID is invalid.	The value submitted for x_device_id is invalid.
3	301	The device batch ID is invalid.	The value submitted for x_device_batch_id is invalid.
3	303	The device batch is full. Please close the batch.	The current device batch must be closed manually from the POS device.
3	304	The original transaction is in a closed batch.	The original transaction has been settled and cannot be reversed.
3	305	The merchant is configured for auto-close.	This merchant is configured for auto-close and cannot manually close batches.
3	306	The batch is already closed.	The batch is already closed.
1	307	The reversal was processed successfully.	The reversal was processed successfully.
1	308	Original transaction for reversal not found.	The transaction submitted for reversal was not found.
3	309	The device has been disabled.	The device has been disabled.
1	310	This transaction has already been voided.	This transaction has already been voided.
1	311	This transaction has already been captured	This transaction has already been captured.
3	312	The specified security code was invalid.	The customer entered the wrong security code. A new security code will be generated, and the customer will be prompted to try again until successful.
3	313	The customer requested a new security code.	The customer requested a new security code. A new security code will be generated, and the customer will be prompted to try again until successful.
2	315	The credit card number is invalid.	This is a processor-issued decline.
2	316	The credit card expiration date is invalid.	This is a processor-issued decline.
2	317	The credit card has expired.	This is a processor-issued decline.
2	318	A duplicate transaction has been submitted.	This is a processor-issued decline.
2	319	The transaction cannot be found.	This is a processor-issued decline.

Response Example for Partial Authorization Transactions

If a split-tender ID is passed in, the response includes each of the following fields. Each field holds the current transaction followed by all transactions associated with the given split-tender ID, in order from oldest to newest. A pipe (|) character separates each value. All parameters hold the same number of values. [Example 9](#) uses simulated values:

Example 9 Partial Authorization Transaction Fields

x_response_code	1 1
x_response_reason_code	1 1
x_response_reason_text	This transaction has been approved. This transaction has been approved.
x_avs_code	Y Y
x_auth_code	C1RR33 04OSH9
x_trans_id	2147801919 2147801918
x_method	CC CC
x_card_type	American Express American Express
x_prepaid_balance_on_card	0.00
x_prepaid_requested_amount	7.53
x_account_number	XXXX0002 XXXX0002
x_cv2_resp_code	
x_cavv_response	2
x_amount	8.0000 1.23

Testing Your Integration

We recommend that you thoroughly test your integration before going live. You can do this by sending transactions to our sandbox, which simulates a production environment. Transactions sent to the sandbox are not forwarded to the payment processing network. You can submit certain values to trigger certain responses. For more information, see the Authorize.Net Testing Guide:

http://developer.authorize.net/hello_world/testing_guide/

Fields by Transaction Type

This appendix lists API fields that should be submitted for each transaction type supported for SIM. It is divided into the following sections:

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

Minimum Required Fields

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

Table 21 Minimum Required Fields

Type of Information	Authorization and Capture	Authorization Only	Prior Authorization and Capture*	Credit *	Void*
Merchant	x_login	x_login	—	—	—
Fingerprint	x_fp_hash	x_fp_hash	—	—	—
	x_fp_sequence	x_fp_sequence			
	x_fp_timestamp	x_fp_timestamp			
Transaction	x_type = AUTH_CAPTURE	x_type = AUTH_ONLY	—	—	—
Payment	x_amount	x_amount	—	—	—
Payment Form Configuration	x_show_form = PAYMENT_FORM	x_show_form = PAYMENT_FORM	—	—	—

* For Prior Authorization and Capture, Credit, and Void transactions, we recommend that the merchant process the transactions by logging on to the merchant interface directly or by using a desktop application that uses AIM.

Required Fields for Advanced SIM Features

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

Table 22 Required Fields for Advanced SIM Features

Type of Information	Authorization and Capture	Authorization and Capture	Prior Authorization and Capture*	Credit *	Void*
Itemized Order	x_line_item	x_line_item	—	—	—
Relay Response Configuration	x_relay_response = TRUE x_relay_url	x_relay_response = TRUE x_relay_url	—	—	—
Advanced Fraud Detection Suite™ (AFDS)	x_customer_ip (required only when the merchant is using customer IP-based AFDS filters)	x_customer_ip (required only when the merchant is using customer IP-based AFDS filters)	—	—	—

* For Prior Authorization and Capture, Credit, and Void transactions, we recommend that the merchant process the transactions by logging on to the merchant interface directly or by using a desktop application that uses AIM.

Best Practice Fields

The following table provides a quick reference of additional API fields that we recommend be submitted per transaction in order to maintain a strong connection.

Table 23 Best Practice Fields

Type of Information	Authorization and Capture	Authorization Only	Prior Authorization and Capture*	Credit *	Void*
Transaction	x_version = 3.1	x_version = 3.1	—	—	—
Payment Form Configuration	x_header_html_payment_ form x_footer_html_payment_ form	x_header_html_payment_ form x_footer_html_payment_ form	—	—	—

Table 23 Best Practice Fields (Continued)

Type of Information	Authorization and Capture	Authorization Only	Prior Authorization and Capture*	Credit *	Void*
Receipt Page Configuration	x_receipt_link_method	x_receipt_link_method	—	—	—
	x_header_html_receipt	x_header_html_receipt			
	x_footer_html_receipt	x_footer_html_receipt			

* For Prior Authorization and Capture, Credit, and Void transactions, we recommend that the merchant process the transactions by logging on to the merchant interface directly or by using a desktop application that uses AIM.

Alphabetized List of API Fields

Table 24 Alphabetized List of API Fields

Field Name	Description
x_address	<p>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 Fields," page 41.</p> <p>Value: The customer's billing address.</p> <p>Format: 60-character maximum (no symbols).</p> <p>Notes: Required if the merchant would like to use the Address Verification Service security feature.</p> <p>For more information on AVS, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p> <p>Required for zero dollar authorizations for Visa verification transactions.</p>
x_amount	<p>Required if x_type = AUTH_CAPTURE, AUTH_ONLY, or CREDIT.</p> <p>Value: The amount of the transaction.</p> <p>Format: 15-digit maximum with a decimal point (no dollar symbol).</p> <p>For example, 8.95.</p> <p>Notes: The total amount to be charged or credited <i>including</i> tax, shipping, and any other charges. The amount can either be hard coded or posted to a script.</p>
x_background_url	<p>Optional.</p> <p>Value: The URL of the merchant's background image.</p> <p>Notes: The image referenced by this URL is displayed as the background on the hosted payment form or receipt page.</p> <p>Background images must be uploaded to the payment gateway server. See "Logos and Background Images for the Hosted Payment Form," page 49, for more information on how to upload images.</p>
x_cancel_url	<p>Optional.</p> <p>Value: The URL to which the gateway redirects the customer's browser when the customer clicks the cancel link.</p> <p>Notes: An API parameter only and not available as a setting in the Merchant Interface.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_cancel_url_text	Optional. Value: This is custom text for the Cancel button. Notes: An API parameter only and not available as a setting in the Merchant Interface.
x_card_num	This field applies only if you use the Direct Post Method. For more information see " Direct Post Method (DPM) ." Value: The customer's full credit card number. Notes: Should not be used with the hosted payment form.
x_city	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 Fields ," page 41. Value: The city of the customer's billing address. Format: 40-character maximum (no symbols).
x_color_background	Optional. Value: The hosted payment form or receipt page background color. Format: Any valid HTML color name or color hex code. Notes: This field is common to the hosted payment form and receipt page. The value in this field sets the background color for both.
x_color_link	Optional. Value: The hosted payment form and receipt page hyperlink color. Format: Any valid HTML color name or color hex code. Notes: This field is common to the hosted payment form and receipt page. The value in this field sets the color of the HTML links for both.
x_color_text	Optional. Value: The hosted payment form and receipt page text color. Format: Any valid HTML color name or color hex code. Notes: This field is common to the hosted payment form and receipt page. The value in this field will set the color of the text for both.
x_company	Optional. Value: The company associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_country	Required when you use a European payment processor. Value: The country of the customer's billing address. Format: 60-character maximum (no symbols).

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_currency_code	<p>Optional.</p> <p>Value: USD, CAD, GBP, AUD, or NZD.</p> <p>Format: 3-character string.</p> <p>Notes: If you do not submit this field, the payment gateway will use the currency selected by the merchant's payment processor. Setting this field to a currency that is not supported by the payment processor results in an error.</p>
x_cust_id	<p>Optional.</p> <p>Value: The merchant-defined customer ID.</p> <p>Format: 20-character maximum (no symbols).</p> <p>Notes: The unique identifier to represent the customer associated with the transaction. The customer ID must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>For this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>
x_description	<p>Optional.</p> <p>Value: The transaction description.</p> <p>Format: 255-character maximum (no symbols).</p> <p>Notes: The description must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>For this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>
x_discount	<p>Value: Enables you to include a discount amount.</p> <p>Format: Enables you to include a discount amount.</p> <p>Value: Positive numeric string with a maximum of 20 characters.</p>
x_duplicate_window	<p>Optional.</p> <p>Value: The period (window) of time after a transaction is submitted during which a duplicate transaction cannot be submitted.</p> <p>Format: Any value from 0 through 28800 (no commas).</p> <p>Notes: Indicates in seconds the period of time after a transaction is submitted during which the payment gateway checks for a duplicate transaction. The maximum time allowed is 8 hours (28800 seconds).</p> <p>If a value less than 0 is sent, the payment gateway defaults to 0 seconds. If a value greater than 28800 is sent, the payment gateway defaults to 28800. If no value is sent, the payment gateway defaults to 2 minutes (120 seconds).</p> <p>If this field is present in the request with or without a value, an enhanced duplicate transaction response is sent. See "Response for Duplicate Transactions," page 75, for more information.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_duty	<p>Value: The valid duty amount OR delimited duty information.</p> <p>Format: When you submit delimited duty information, field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total duty amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited duty information. This information includes:</p> <ul style="list-style-type: none"> ■ duty item name< > <p>Value: The duty item name.</p> ■ duty description< > <p>Value: The duty item description.</p> ■ duty amount <p>Value: The duty amount. The total amount of the transaction in x_amount must <i>include</i> this amount.</p> <p>Format: The dollar sign (\$) is not allowed when you submit delimited information.</p> <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_duty" VALUE="Duty1< >export< >15.00"></pre>
x_email	<p>Required when you use a European payment processor.</p> <p>Value: The customer's valid email address.</p> <p>Format: 255-character maximum.</p> <p>For example, janedoe@customer.com.</p> <p>Notes: The email address to which the customer's copy of the email receipt is sent when Email Receipts is configured in the Merchant Interface. The email is sent to the customer only if the email address format is valid.</p> <p>For more information about Email Receipts, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>
x_email_customer	<p>Optional.</p> <p>Value: The customer email receipt status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0.</p> <p>Notes: Indicates whether an email receipt should be sent to the customer.</p> <p>If set to TRUE, the payment gateway sends an email to the customer after the transaction is processed using the customer email address submitted with the transaction. If FALSE, no email is sent to the customer.</p> <p>If no value is submitted, the payment gateway looks up the configuration in the Merchant Interface and sends an email only if the merchant has enabled the setting. If this field is not submitted, and the setting is disabled in the Merchant Interface, no email is sent.</p> <p>For more information about configuring Email Receipts in the Merchant Interface, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_fax	Optional. Value: The fax number associated with the customer's billing address. Format: 25-digit maximum (no letters). For example, (123)123-1234.
x_first_name	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 Fields," page 41 . Value: The first name associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_footer_email_receipt	Optional. Value: The email receipt footer. Format: Plain text. Notes: This text appears as the footer on the email receipt sent to the customer.
x_footer_html_payment_form	Optional. Value: The hosted payment form footer. Format: Plain text. Avoid using double quotes. Notes: The text submitted in this field is displayed as the footer on the hosted payment form.
x_footer2_html_payment_form	Optional. Format: Plain text. Avoid using double quotes. Notes: Same as x_footer_html_payment_form , except that it appears at the very top of the page, above the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.
x_footer_html_receipt	Optional. Value: The hosted receipt page footer. Format: Plain text. Avoid using double quotes. Notes: The text submitted in this field is displayed at the bottom of the hosted receipt page.
x_footer2_html_receipt	Optional. Format: Plain text. Avoid using double quotes. Notes: Same as x_footer_html_receipt , except that it is displayed at the bottom of the page below the box. This is an API parameter only; it is not available as a setting in the merchant interface. This is shown for approvals, declines, and errors.

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_fp_hash	<p>Required.</p> <p>Value: The unique transaction fingerprint.</p> <p>Notes: The fingerprint is generated using the HMAC-SHA512 hashing algorithm on the following field values:</p> <p>API login ID (x_login)</p> <p>The sequence number of the transaction (x_fp_sequence)</p> <p>The timestamp of the sequence number creation (x_fp_timestamp)</p> <p>Amount (x_amount)</p> <p>Currency code, if submitted (x_currency_code)</p> <p>Field values are concatenated and separated by a caret (^).</p>
x_fp_sequence	<p>Required</p> <p>Value: The merchant-assigned sequence number for the transaction.</p> <p>Format: Numeric.</p> <p>Notes: The sequence number can be a merchant-assigned value, such as an invoice number or any randomly generated number.</p>
x_fp_timestamp	<p>Required</p> <p>Value: The timestamp at the time of fingerprint generation.</p> <p>Format: UTC time in seconds since January 1, 1970.</p> <p>Notes: Coordinated Universal Time (UTC) is an international atomic standard of time (sometimes referred to as GMT). Using a local time zone timestamp causes fingerprint authentication to fail.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_freight	<p>Value: The valid freight amount OR delimited freight information.</p> <p>Format: When you submit delimited freight information, field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total freight amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited freight information. This information includes:</p> <p>Delimited freight information fields include:</p> <ul style="list-style-type: none"> ■ freight item name< > <p>Value: The freight item name.</p> ■ freight item name< > <p>Value: The freight item description.</p> ■ freight amount <p>Value: The freight amount. The total amount of the transaction in x_amount must <i>include</i> this amount.</p> <p>Format: The dollar sign (\$) is not allowed when submitting delimited information.</p> <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_freight" VALUE="Freight1< >ground overnight< >12.95></pre>
x_giftwrap	<p>Value: Enables you to charge for gift wrap.</p> <p>Format: Numeric string with a maximum of 20 characters.</p> <p>Value: Only used for Visa Checkout.</p>
x_header_email_receipt	<p>Optional.</p> <p>Value: The email receipt header.</p> <p>Format: Plain text.</p> <p>Notes: This text appears as the header of the email receipt sent to the customer.</p>
x_header_html_payment_form	<p>Optional.</p> <p>Value: The hosted payment form header.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text submitted in this field is displayed as the header on the hosted payment form.</p>
x_header2_html_payment_form	<p>Notes: Same as x_header_html_payment_form except that it appears at the top of the page, above the box. It is an API parameter only; it is not available as a setting in the Merchant Interface.</p>
x_header_html_receipt	<p>Optional.</p> <p>Value: The hosted receipt page header.</p> <p>Format: Plain text. Avoid using double quotes.</p> <p>Notes: The text submitted in this field is displayed at the top of the hosted receipt page.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_header2_html_receipt	Same as x_header_html_receipt except it appears at the top of the page above the box. This is an API parameter only; it is not available as a setting in the merchant interface. This is shown for approvals, declines, and errors.
x_invoice_num	Optional. Value: The merchant-assigned invoice number for the transaction. Format: 20-character maximum (no symbols). Notes: The invoice number must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function. For this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.
x_last_name	Required when using a European payment processor. If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41 . Value: The last name associated with the customer's billing address. Format: 50-character maximum (no symbols).
x_line_item	Value: Itemized Order Information. Format: Any string. Line item values must be delimited by a bracketed pipe < > Child elements include, in this order: <ul style="list-style-type: none"> ■ item ID< > Format: 31-character maximum. ■ item name< > Format: 31-character maximum. ■ item description< > Format: 255-character maximum. ■ item quantity< > Format: Up to 2 decimal places. Must be a positive number. ■ item price (unit cost)< > Format: Up to 2 decimal places. Must be a positive number. Notes: Cost of an item per unit, excluding tax, freight, and duty. The dollar sign (\$) is not allowed when submitting delimited information. ■ item taxable Format: TRUE, FALSE,T, F,YES, NO,Y, N,1, 0 Notes: This field can be submitted more than once.

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_login	<p>Required.</p> <p>value: The merchant's unique API login ID.</p> <p>Format: 20-character maximum.</p> <p>Notes: The merchant API login ID is provided in the Merchant Interface and must be stored securely.</p> <p>The API login ID and transaction fingerprint together provide the merchant authentication required for access to the payment gateway.</p> <p>For more information, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>
x_logo_url	<p>Optional.</p> <p>Value: The URL of the merchant's logo.</p> <p>Notes: The image referenced by this URL is displayed in the header or footer of the hosted payment form and receipt page.</p> <p>Logo images must be uploaded to the payment gateway server. See "Logos and Background Images for the Hosted Payment Form," page 49, for more information on how to upload images.</p>
x_method	<p>Optional.</p> <p>Value: The payment method.</p> <p>Format: CC or ECHECK.</p> <p>Notes: The method of payment for the transaction, CC (credit card) or ECHECK (electronic check). If you specify a payment method, only that option will be available on the payment form.</p> <p>For more information about eCheck.Net transaction requirements, see the <i>eCheck.Net Developer Guide</i>: http://developer.authorize.net/guides/echeck.pdf</p>
x_misc	<p>Notes: Enables you to allow miscellaneous charges.</p> <p>Format: Numeric string with a maximum of 20 characters.</p> <p>Notes: Used only for Visa Checkout.</p>
x_phone	<p>Optional.</p> <p>Value: The phone number associated with the customer's billing address.</p> <p>Format: 25-digit maximum (no letters). For example, (123)123-1234.</p>
x_po_num	<p>Required only if your payment processor is EVO and you submit Level 2 data.</p> <p>Value: The merchant-assigned purchase order number.</p> <p>Format: 25-character maximum (no symbols).</p> <p>Notes: The purchase order number must be created dynamically on the merchant server or provided per transaction. The payment gateway does not perform this function.</p> <p>For this field to be included on the hosted payment form, the View attribute for the field must be configured in the Merchant Interface payment form settings.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_promo_code	<p>Notes: Enables you to accept a promo code.</p> <p>Format: Alphanumeric string with a maximum of 100 characters.</p> <p>Value: Used only for Visa Checkout.</p>
x_receipt_link_method	<p>Optional.</p> <p>Value: The type of link back to the merchant's web site from the hosted receipt page.</p> <p>Format: LINK, POST, or GET.</p> <p>LINK creates a hyperlink.</p> <p>GET creates a button and returns transaction information in the receipt link URL.</p> <p>POST creates a button and returns transaction information as an HTML Form POST.</p>
x_receipt_link_text	<p>Optional.</p> <p>Value: The text of the link or button that directs the customer back to the merchant's web site.</p> <p>Format: 50-character maximum.</p> <p>Notes: If the x_receipt_link_method is LINK, the value in this field becomes a hyperlinked text on the hosted receipt page. If the x_receipt_link_method is GET or POST the value in this field becomes the text of a submit button. An HTML form is created in the receipt page that has hidden fields containing the results of the transaction processed.</p>
x_receipt_link_url	<p>Optional.</p> <p>Value: The URL of the link or button that directs the customer back to the merchant's web site.</p> <p>Notes: To be accepted as valid by the payment gateway, the URL must be configured in the Merchant Interface.</p> <p>If the receipt link method is LINK, the URL specified becomes the href value of the hyperlinked text. If the receipt link method is GET or POST, the URL becomes the action of the HTML form.</p>
x_recurring_billing	<p>Optional.</p> <p>Value: The recurring billing status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0.</p> <p>Notes: Marker used by merchant account providers to identify transactions that originate from merchant-hosted recurring billing applications. This value is not affiliated with automated recurring billing.</p>
x_relay_always	<p>This field should always be set to true when you use the Direct Post Method. For more information, see "Direct Post Method (DPM)."</p> <p>Value: Requests a relay response even for partial authorizations and in case of errors.</p> <p>Format: TRUE, FALSE.</p> <p>Notes: This field instructs the payment gateway to return a relay response regardless of any declines, errors, or partial authorizations.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_relay_response	<p>Optional when using the hosted form, required for the Direct Post Method.</p> <p>Value: The request for a relay response.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0.</p> <p>Notes: This field instructs the payment gateway to return transaction results to the merchant by means of an HTML form POST to the merchant's web server for a relay response.</p>
x_relay_URL	<p>Optional when using the hosted form, required for the Direct Post Method.</p> <p>Value: The URL on the merchant's web site to which the payment gateway should post transaction results for a relay response.</p> <p>Format: Any valid URL.</p> <p>Including name/value pairs in the URL (anything after a question mark (?)) is not recommended.</p> <p>Notes: If this field is submitted, the payment gateway validates the URL value against the Relay Response URL configured in the Merchant Interface. If the URL submitted does not match the URL configured in the Merchant Interface, the transaction is rejected. If no value is submitted in the HTML Form POST, the payment gateway posts the transaction results to the URL configured in the Merchant Interface.</p>
x_rename	<p>Optional.</p> <p>Value: A request to rename a field.</p> <p>Format: Field name on the payment form, new field name.</p> <p>Notes: Use this variable to replace a field name on a payment form. It does not rename the original field, it only changes the value displayed on the payment form.</p> <p>See ",," page 51, for more information.</p>
x_ship_to_address	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The customer's shipping address.</p> <p>Format: 60-character maximum (no symbols).</p>
x_ship_to_company	<p>Optional.</p> <p>Value: The company associated with the customer's shipping address.</p> <p>Format: 50-character maximum (no symbols).</p>
x_ship_to_country	<p>Optional.</p> <p>Value: The country of the customer's shipping address.</p> <p>Format: 60-character maximum (no symbols).</p>
x_ship_to_city	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The city of the customer's shipping address.</p> <p>Format: 40-character maximum (no symbols).</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_ship_to_first_name	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The first name associated with the customer's shipping address.</p> <p>Format: 50-character maximum (no symbols).</p>
x_ship_to_last_name	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The last name associated with the customer's shipping address.</p> <p>Format: 50-character maximum (no symbols)</p>
x_ship_to_state	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The state of the customer's shipping address.</p> <p>Format: 40-character maximum (no symbols) or a valid 2-character state code.</p>
x_ship_to_zip	<p>If your payment processor is EVO and you submit this field, other fields are required. See "EVO Billing and Shipping Fields," page 41.</p> <p>Value: The ZIP code of the customer's shipping address.</p> <p>Format: 20-character maximum (no symbols).</p>
x_show_form	<p>Required for the hosted payment form. This field cannot be set when you use the Direct Post Method.</p> <p>Value: The payment form request.</p> <p>Format: PAYMENT_FORM.</p> <p>Notes: The show form field indicates that the merchant wishes to use the payment gateway hosted payment form to collect payment data.</p>
x_state	<p>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 Fields," page 41.</p> <p>Value: The state of the customer's billing address.</p> <p>Format: 40-character maximum (no symbols) or a valid 2-character state code.</p>
x_subtotal	<p>Value: If this field is not submitted, the subtotal is calculated by subtracting all charges and discounts from the amount field.</p> <p>If this field is submitted, it is accepted and shown as is.</p> <p>Format: Numeric string with a maximum of 20 characters.</p> <p>Notes: Used only with Visa Checkout.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_tax	<p>Value: The valid tax amount OR delimited tax information.</p> <p>Format: When you submit delimited tax information, the field values must be delimited by a bracketed pipe < >.</p> <p>Notes: The value of this field is typically the total tax amount. However, if you are submitting this information in an HTML Form POST, you can submit delimited tax information. This information includes:</p> <ul style="list-style-type: none"> ■ tax item name< > ■ tax description< > ■ tax amount <p>Format: The dollar sign (\$) is not allowed when you submit delimited information.</p> <p>Note: The total amount of the transaction in x_amount must include this amount.</p> <p>Example:</p> <pre><INPUT TYPE="HIDDEN" name="x_tax" VALUE="Tax1< >state tax< >0.0625"></pre>
x_tax_exempt	<p>Optional.</p> <p>Value: The tax exempt status.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0.</p> <p>Notes: Indicates whether the transaction is tax exempt.</p> <p>The total amount of the transaction in x_amount must <i>include</i> this amount.</p>
x_test_request	<p>Optional.</p> <p>Value: The request to process test transactions.</p> <p>Format: TRUE, FALSE, T, F, YES, NO, Y, N, 1, 0.</p> <p>Notes: Indicates whether the transaction should be processed as a test transaction.</p> <p>See "Testing Your Integration," page 90, for more information.</p>
x_type	<p>Optional.</p> <p>Value: The type of credit card transaction.</p> <p>Format: AUTH_CAPTURE (default), AUTH_ONLY.</p> <p>Notes: If the value submitted does not match a supported value, the transaction is rejected. If no value is submitted in this field, the payment gateway processes the transaction as an AUTH_CAPTURE.</p> <p>For transaction types CREDIT, PRIOR_AUTH_CAPTURE, and VOID, we recommend that the merchant process the transactions by logging on to the merchant interface directly or by using a desktop application that uses AIM.</p>

Table 24 Alphabetized List of API Fields (Continued)

Field Name	Description
x_zip	<p>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 Fields," page 41.</p> <p>Required when the merchant uses the Address Verification Service security feature.</p> <p>Required for zero dollar authorizations for Visa verification transactions.</p> <p>Value: The ZIP code of the customer's billing address.</p> <p>Format: 20-character maximum (no symbols).</p> <p>Notes: For more information on AVS, see the <i>Merchant Integration Guide</i>: http://www.authorize.net/support/merchant/</p>

Direct Post Method (DPM)



Important

We now offer [Accept.js](#), a JavaScript replacement for Direct Post Method (DPM). Accept.js is a modern implementation that doesn't require a post-back. With Accept.js, developers have control over the user experience without sending PCI data through their servers. Accept.js can be used like any other payment type in the [Authorize.Net API](#).

We strongly encourage new development using Accept.js instead of DPM, and developers with existing DPM implementations should consider moving to Accept.js.

There is an alternate method of implementing a payment form, called the Direct Post Method. This method is similar to SIM, with a few key differences.

Differences From SIM

When you use SIM, you connect to Authorize.Net's hosted payment form. When you use DPM, you host your own payment form, giving you more control over its appearance. The payment information is then directly posted to the payment gateway so that it is never stored on the merchant's server.

There are a few differences in the API fields that you submit, depending on whether you use SIM or DPM:

- With SIM, you use **x_show_form**. With DPM, you do not.
- With DPM, you also use **x_card_num** and **x_exp_date**.
- With DPM, **x_relay_always** should be set to true. As a result, an error never displays on the payment form; instead, the relay response is triggered.

When you use Relay/Response URL whitelisting, and the URL specified does not match the value of **x_relay_url**, you do not receive an error. Instead, the user is redirected to the default relay URL in that whitelist. For more information, see "[Whitelisting](#)," page 61.

Relay Response

When you use DPM, Authorize.Net sends a POST of the transaction result to the relay URL. Your relay response page then redirects the client's browser to the merchant's server. Therefore, the URL in the client's browser shows the merchant's server and not Authorize.Net's server.

Authorize.Net expects a return from this HTTP POST and displays the result:

- On failure (an HTTP response code other than 200-OK), the Authorize.Net server returns an error message.
- On success, the Authorize.Net server returns the code generated by the merchant's relay URL, which forces the redirection of the client's browser to the merchant's server. This redirect uses JavaScript if available on the client's browser and a *meta refresh* tag if it is not. For example:

```
<html>
<head>
  <script type="text/javascript" charset="utf-8">
    window.location='http://YOUR_SERVER.COM/receipt.jsp';
  </script>
  <noscript>
    <meta http-equiv="refresh" content="1;url=http://
YOURSERVER.COM/receipt.jsp">
  </noscript>
</head>
<body></body>
</html>
```

Conceptual Overview

The following diagram and table describe the architecture and flow of control for a DPM transaction:

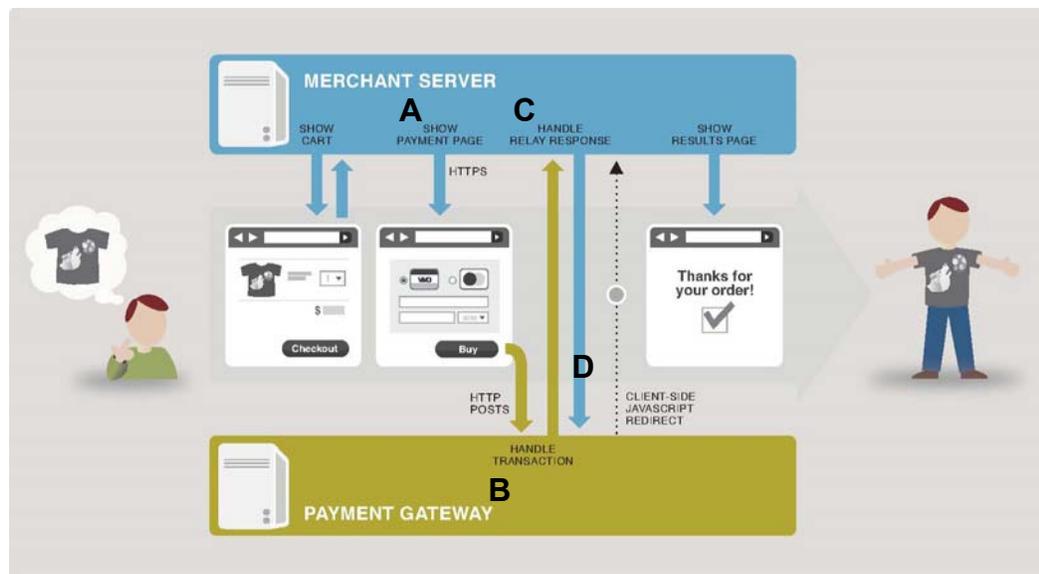


Table 25 Transaction Flow Diagram

Diagram Label	Description
A	When a customer submits an order on the shopping cart, the merchant's server generates a payment form to collect the customer's payment and shipping information.
B	<p>The payment form posts directly to Authorize.Net, bypassing the merchant server. This post can include both hidden (merchant-supplied) and customer data fields. The developer can use any of the API form fields and can create merchant-defined data fields. "Merchant-Defined Fields," page 50, identifies and defines common form fields and specifies the constraints on merchant-defined fields.</p> <p>The posted form includes an x_relay_url value containing the URL to which Authorize.Net posts transaction results upon completion.</p> <p>This is not a typical relay URL; it contains no content for display in the client's browser. Instead, it returns code that redirects the client's browser to a URL on the merchant's server. The redirect maintains the merchant server's URL in the client's browser address bar throughout the transaction.</p>
C/D	When the merchant's server receives the HTTP POST from Authorize.Net, it validates the hash values and logs the order. The content of the relay page that is used with DPM immediately redirects the user to another URL on the merchant's site. The redirect keeps the URL in the client's browser pointed to the merchant's server. The redirect should also contain enough information about the transaction so that the merchant's server can display something sensible to the user.

Address and Card Code Verification

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

<http://www.authorize.net/support/merchant/>

Visa Checkout

Visa Checkout is a digital payment solution that stores customer payment information. Customers who enroll in Visa Checkout can simply log in instead of re-entering their payment information every time they check out. You can offer Visa Checkout as a payment option on the hosted payment form.

Enabling Visa Checkout for Your Account

You must enable Visa Checkout for an account before you can add it as a payment option on the hosted payment form.

To enable Visa Checkout for an account:

- Step 1** Navigate to the **Accounts** section of the Merchant Interface.
- Step 2** Click **Digital Payment Solutions**.
- Step 3** In the Visa Checkout section, click **Sign Up**. If the Sign Up option is not present, call customer support.
- Step 4** Review the account information on the next screen. Click **Edit** to edit the information in the My Information and Business Information sections. To edit the information in the Additional Business Information section, type directly into the text fields.
- Step 5** When you are finished editing the information on this page, click **Additional Services Addendum** and read it.
- Step 6** Check the box to confirm that you have read the Additional Services Addendum.
- Step 7** Click **Agree**. When you return to the Digital Payment Solutions page, the status of Visa Checkout will be Pending while your request is being processed. When the request is successfully processed, the status will change to Enabled. To continue the setup process, click **Enabled**. The next page contains a Visa Checkout API key and a link to payment form settings. The API key is not relevant to the hosted payment form and you can ignore it if you are using the SIM API.
- Step 8** Below the API key section, click **Visa Checkout specific payment form settings**. You can also navigate to this section by choosing **Accounts > Settings > Transaction**

Format Settings > Transaction Submission Settings > Payment Form > Visa Checkout Settings. The Visa Checkout payment Form Setting page is displayed.

- Step 9** Enable Visa Checkout on your payment form by setting the Visa Checkout Payment Option slider to **Yes**.
- Step 10** To collect shipping information in Visa Checkout transactions, set the Shipping Information slider to **Yes**.

Visa Checkout API Fields

The following fields can be called with the request for the payment form.

Table 26 Visa Checkout API Fields

Field Name	Value	Format
x_subtotal	If this field is not submitted, the subtotal is calculated by subtracting all charges and discounts from the amount field. If this field is submitted, it is accepted and shown as is.	Numeric string with a maximum of 20 characters.
x_giftwrap	Enables you to charge for gift wrap.	Numeric string with a maximum of 20 characters.
x_discount	Enables you to include a discount amount.	Positive numeric string with a maximum of 20 characters.
x_misc	Enables you to allow miscellaneous charges.	Numeric string with a maximum of 20 characters.
x_promo_code	Enables you to accept a promo code.	Alphanumeric string with a maximum of 100 characters.



Note

The Visa Checkout API fields are not included in Silent POST or relay response.

Browser Support

Visa Checkout for Authorize.Net is supported on:

- Internet Explorer versions 9 and later
- Firefox versions 32 and later
- Chrome versions 31 and later

Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

address verification service [14](#)
 Advanced Integration Method [11](#)
 AIM [11](#)
 attributes for additional fields [35](#)
 authorization only [25](#)
 AVS [14](#)

B

background images [59](#)
 buttons, displaying
 sample code [43](#)

C

capture only [26](#)
 card code verification [14](#)
 customizing payment form [35](#)

D

direct post method [12](#)
 DPM [12](#)
 duplicate transactions [65](#)

E

eCheck.net [15](#)
 email receipts [62](#)

F

features [13](#)

fields

 merchant defined [50](#)
 optional [64](#)

form fields

 minimum [31](#)

H

hashing algorithm [29](#)
 hosted payment form [31](#)
 background images [49](#)
 logos [49](#)
 minimum fields [31](#)
 size in pixels [50](#)
 hosted receipt page
 customizing [52](#)
 HTML post URL [28](#)

I

images
 displaying on hosted payment form [49](#)
 integration settings [13](#)
 itemized order information [14](#)

L

logos [59](#)

M

merchant interface
 using [27](#)
 minimum requirements [12](#)
 MOTO transactions [24](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

O

optional fields [64](#)

orders

 itemizing [14](#)

P

payment form

 configurable fields [35](#)

 configuring form fields [35](#)

 customizing [43](#)

payment method [64](#)

ports [61](#)

prior authorization and capture [25](#)

R

receipt link URL [53](#)

receipt page [14](#)

 customizing [52](#)

receipt method [53](#)

refunds [26](#)

relay response [59](#)

required fields

 minimum [31](#)

S

secure hosted payment form [31](#)

SIM

 features [13](#)

SSL [11](#)

T

test transactions [65](#)

transaction key [30](#)

transaction settings [13](#)

transaction types

 authorization and capture [24](#)

 authorization only [25](#)

 credit [26](#)

 prior authorization and capture [25](#)

 void [26](#)

transactions

 authenticating [29](#)

 canceling [26](#)

 duplicate [65](#)

 electronic check [15](#)

 MOTO [24](#)

 posting [28](#)

 security [11](#)

 testing [65](#)

 types [24](#)

transction types

 capture only [26](#)

X

x_address [37](#)

x_amount [31](#)

x_background_url [45](#)

x_cancel_url [45](#)

x_cancel_url_text [45](#)

x_city [37](#)

x_color_background [44](#)

x_color_link [44](#)

x_color_text [44](#)

x_company [37](#)

x_country [38](#)

x_currency_code [96](#)

x_cust_id [38](#)

x_description [36](#)

x_duty [40](#)

x_email [38, 62](#)

x_email_customer [62](#)

x_fax [38](#)

x_first_name [36](#)

x_footer2_html_payment_form [44](#)

x_footer2_html_receipt [58](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

x_footer_email_receipt **62**
x_footer_html_payment_form **44**
x_footer_html_receipt **58**
x_fp_hash **29**
x_fp_sequence **29**
x_fp_timestamp **29**
x_freight **40**
x_header2_html_payment_form **44**
x_header2_html_receipt **58**
x_header_email_receipt **62**
x_header_html_payment_form **44**
x_header_html_receipt **58**
x_invoice_num **36**
x_last_name **37**
x_logo_url **45**
x_phone **38**
x_po_num **41**
x_receipt_link_method **53**
x_receipt_link_text **54**
x_receipt_link_url **54**
x_recurring_billing **36**
x_relay_always **60**
x_relay_response **60**
x_relay_url **60**
x_return_policy_url **44**
x_ship_to_address **39**
x_ship_to_city **39**
x_ship_to_company **39**
x_ship_to_country **39**
x_ship_to_first_name **38**
x_ship_to_last_name **38**
x_ship_to_state **39**
x_ship_to_zip **39**
x_show_form **31**
x_state **37**
x_tax **39**
x_tax_exempt **40**
x_test_request **65**
x_type **31**
x_version **64**
x_zip **37**