Advanced Integration Method (AIM) Implementation Guide Card-Not-Present Transactions Version 1.0

Table of Contents

Introduction

Payment gateways facilitate electronic commerce by enabling merchants to accept credit cards and electronic checks as methods of payment for goods and services sold online. The gateway acts as a bridge between the merchant's Website and the financial institutions that process payment transactions. Payment data is collected online from the shopper and submitted to the gateway for real-time authorization.

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

Note: The payment gateway is targeted towards merchants that process Card-Not-Present transactions. In a Card-Not-Present transaction, the merchant and the shopper are not in the same physical location and the customer usually calls in the payment data or keys in the details of the credit card on a Website. All e-commerce and mail/telephone orders are Card-Not-Present transactions.

The gateway also supports electronic check transactions. Merchants can collect customer bank account numbers and routing numbers to pay for purchases.

This document describes how transactions can be submitted to the gateway for real-time processing using Advanced Integration Method (AIM).

AIM is the recommended integration method for merchants who have the capability to initate both client and server side SSL connections. This method offers the merchant a high degree of security and control because transaction data is submitted to the gateway over a secure server-to-server connection that is initiated by the merchant server. Since the merchant server will receive a response directly from the gateway, the merchant has more control over the response to the end customer.

Advanced Integration Method (AIM)

What is AIM?

AIM is the recommended method of submitting transactions to the payment gateway. This method allows a merchant's server to securely connect directly to the payment gateway to submit transaction data. The merchant retains full control of the payment data collection and the user experience. This method requires merchants to be able to initiate and manage secure Internet connections.

How Does AIM Work?

When using AIM, transactions flow in the following way:

1. The Customer's browser connects securely to the Merchant's server to transmit payment information.
2. The Merchant's server initiates a secure connection to the payment gateway and then initiates an HTTPS post of the transaction data to the gateway server.
3. The payment gateway receives and processes the transaction data.
4. The payment gateway then generates and submits the transaction response to the Merchant's server.
5. The Merchant's server receives and processes the response.
6. Finally, the Merchant's server communicates the success or failure of the authorization to the Customer's browser.

What is Required to Implement AIM?

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

1. Establish a secure socket connection
2. Provide both server and client side encryption
3. Develop scripts on a Web server for the integration to the gateway (e.g., for submitting transaction data and receiving and translating system responses)
4. Securely store a transaction key to be accessed by the script that submits the transaction to the gateway.

The AIM Application Program Interface (API)

The Standard Transaction Submission API defines how transactions should be submitted to the gateway using AIM. The gateway response API describes the gateway's responses to transactions submitted to the gateway. These APIs are discussed in detail in this document.

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

AIM Implementation

To implement AIM, a developer would design a script that does the following:

1. Securely obtains all of the information needed to process a transaction.
2. Initiates a secure HTTPS form POST from their server to **https://secure.authorize.net/gateway/transact.dll**. Note: Authorize.Net will only accept transactions on port 443. This post will include all system variables mentioned in the tables below (see the following section entitled "Standard Transaction Submission API for AIM").
3. Receives the response from the gateway and processes the response to display the appropriate result to the end user.

Using the Merchant Interface to Configure AIM

Merchants submitting transactions via AIM can configure how the gateway should construct the response back to the merchant server initiating the request.

- By default, the response fields will be *delimited* with a comma. The merchant can override the default separator and specify what character should separate the response fields.
- The response fields will not be *encapsulated* by default. The merchant can configure the encapsulation character. It is recommended that the merchant override the system default and set an encapsulation character.

The delimiting character and the encapsulation character can be set in the Merchant Interface by doing the following:

1. Log in-to the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click *Direct Response* from the Transaction Response section
4. Configure the settings:
    a. Set *Delimited Response* to Yes
    b. Choose the *Default Delimited Separator* from the drop-down box or enter a customized value
    c. Choose the *Field Encapsulation Character* from the drop-down box or enter a customized value
6. Click *Submit* to save changes

Minimum Requirements for AIM

The following is the minimum set of NAME/VALUE pairs that should be submitted to the gateway when using AIM for a credit card transaction.

FIELD NAME	**FIELD VALUE**
x_Version	3.1
x_Delim_Data	True
x_Login	*Your Login ID*
x_Tran_Key	*Transaction key obtained from the Merchant Interface*
x_Amount	*Amount of purchase inclusive of tax*

x_Card_Num	*Customer's card number*
x_Exp_Date	*Customer's card expiration date*
x_Type	*Type of transaction (*AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT, VOID, PRIOR_AUTH_CAPTURE

The following is the minimum set of NAME/VALUE pairs that should be submitted to the gateway when using AIM for an eCheck transaction.

FIELD NAME	**FIELD VALUE**
x_Version	3.1
x_Delim_Data	True
x_Login	*Your Login ID*
x_Tran_Key	*Transaction key obtained from the Merchant Interface*
x_Amount	*Amount of purchase inclusive of tax*
x_Bank_ABA_Code	*ABA routing number*
x_Bank_Acct_Num	*Bank Account Number*
x_Bank_Acct_Type	*Type of Account – Checkings or Savings*
x_Bank_Name	*Name of bank at which account is maintained*
x_Bank_Acct_Name	*Name underwhich the account is maintained at the bank*
x_Type	*Type of transaction (*AUTH_CAPTURE, CREDIT)

Security Considerations for AIM

Every transaction submitted to the system using AIM should have a transaction key. The transaction key needs to be securely stored on the merchant server and submitted with each transaction. The gateway rejects all transactions that do not have a transaction key or that include an invalid key. The transaction key is generated by the system and can be obtained from Merchant Interface. To obtain the transaction key from the Merchant Interface

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on *Obtain Transaction Key* in the Security section
4. Type in the answer to the secret question configured on setup
5. Click Submit

It is strongly recommended that the merchant periodically change the transaction key. The merchant will have to disable the old key and generate a new key. The old key will be valid for 24 hours before it expires. To disable the old key on the Merchant Interface:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on *Obtain Transaction Key* in the Security section
4. Type in the answer to the secret question configured on setup
5. Check the box that says *Disable Old Key*
6. Click *Submit*

Note: Use only port 443 for AIM information transfers for reasons of security.

Standard Transaction Submission API for AIM

The Standard Transaction Submission API defines the information that can be submitted to the gateway for real-time transaction processing. The API consists of a set of fields that are required for each transaction, and a set of fields that are optional. Under the API, the gateway accepts a NAME/ VALUE pair. The NAME is the field name and indicates to the gateway what information is being submitted. VALUE contains the content of the field.

The following tables contain the data fields that may be submitted to the system with any transaction. The fields are grouped logically in the tables, based on the information submitted. Each table contains the following information:

- *Field* – Name of the parameter that may be submitted on a transaction.
- *Required* – Indicates whether the field is required on a transaction. If *Conditional*, indicates that the field is required based on the existence or value of another field. In cases where a dependency exists, an explanation is provided.
- *Value* – Lists the possible values that may be submitted for the field. In cases where a format is validated, an explanation is provided.
- *Max Length* – Indicates the maximum number of characters that may be supplied for each field.
- *Description* – Provides additional details on how the field is used.

Merchant Account Information

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Login	Required	Varies by merchant	20	Pass the Login ID used to access the Merchant Interface.
x_Tran_Key	Required	Varies by merchant	16	Pass the transaction key obtained from the merchant interface.
x_Version	Optional If no value is specified, the value located in the Transaction Version settings within the Merchant Interface will be used.	2.5, 3.0, 3.1	3	Indicates to the system the set of fields that will be included in the response: • 3.0 is the standard version • 3.1 allows the merchant to utilize the Card Code feature
x_Test_Request	Optional	TRUE, FALSE	5	Indicates whether the transaction should be processed as a test transaction. Please refer to Appendix E for further information on this field.

Gateway Response Configuration

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Delim_Data	Optional	TRUE	5	In order to receive a delimited response from the gateway, this field has to be submitted with a value of TRUE or the merchant has to configure a delimited response through the Merchant Interface.
x_Delim_Char	Optional	Any valid character	1	Character that will be used to separate fields in the transaction response. The system will use the character passed in this field or the value stored in the Merchant Interface if no value is passed. If this field is passed, and the value is null, it will override the value stored in the Merchant Interface and there will be no delimiting character in the transaction response.
x_Encap_Char	Optional	Any valid character	1	Character that will be used to encapsulate the fields in the transaction response. The system will use the character passed in this field or the value stored in the Merchant Interface if no value is passed. If this field is passed, and the value is null, it will override the value stored in the Merchant Interface and there will be no encapsulation character in the transaction response.

Customer Name and Billing Address

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_First_Name	Optional	Any string	50	Contains the first name of the customer associated with the billing address for the transaction.
x_Last_Name	Optional	Any string	50	Contains the last name of the customer associated with the billing address for the transaction.
x_Company	Optional	Any string	50	Contains the company name associated with the billing address for the transaction.
x_Address	Optional	Any string	60	Contains the address of the customer associated with the billing address for the transaction.
x_City	Optional	Any string	40	Contains the city of the customer associated with the billing address for the transaction.
x_State	Optional If passed, the value will be verified.	Any valid two-digit state code or full state name	40	Contains the state of the customer associated with the billing address for the transaction.
x_Zip	Optional	Any string	20	Contains the zip of the customer associated with the billing address for the transaction.
x_Country	Optional If passed, the value will be verified.	Any valid two-digit country code or full country name (spelled in English)	60	Contains the country of the customer associated with the billing address for the transaction.
x_Phone	Optional	Any string Recommended format is (123)123-1234	25	Contains the phone number of the customer associated with the billing address for the transaction.
x_Fax	Optional	Any string Recommended format is (123)123-1234	25	Contains the fax number of the customer associated with the billing address for the transaction.

Additional Customer Data

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Cust_ID	Optional	Any string	20	Unique identifier to represent the customer associated with the transaction.
x_Customer_IP	Optional	Required format is 255.255.255.255. If this value is not passed, it will default to 255.255.255.255	15	IP address of the customer initiating the transaction.
x_Customer_Tax_ID	Optional	9 digits/numbers only	9	Tax ID or SSN of the customer initiating the transaction.

Email Settings

The following fields describe how and when emails will be sent when transactions are processed by the system.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Email	Optional	Any valid email address	255	Email address to which the customer's copy of the confirmation email is sent. No email will be sent to the customer if the email address does not meet standard email format checks.
x_Email_Customer	Optional	TRUE, FALSE If no value is submitted, system will default to the value configured in the Merchant Interface.	5	Indicates whether a confirmation email should be sent to the customer.
x_Merchant_Email	Optional	Any valid email address	255	Email address to which the merchant's copy of the customer confirmation email should be sent. If a value is submitted, an email will be sent to this address as well as the address(es) configured in the Merchant Interface.

Invoice Information

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

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

Customer Shipping Address

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Ship_To_First_Name	Optional	Any string	50	Contains the customer shipping first name.
x_Ship_To_Last_Name	Optional	Any string	50	Contains the customer shipping last name.
x_Ship_To_Company	Optional	Any string	50	Contains the customer shipping company.
x_Ship_To_Address	Optional	Any string	60	Contains the customer shipping address.
x_Ship_To_City	Optional	Any string	40	Contains the customer shipping city.
x_Ship_To_State	Optional If passed, the value will be verified.	Any valid two-digit state code or full state name	40	Contains the customer shipping state.
x_Ship_To_Zip	Optional	Any string	20	Contains the customer shipping zip.
x_Ship_To_Country	Optional If passed, the value will be verified.	Any valid two-digit country code or full country name (spelled in English)	60	Contains the customer shipping country.

Transaction Data

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

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Amount	Required	Any amount	15	Total value to be charged or credited inclusive of tax.
x_Currency_Code	Optional	Valid currency code	3	Currency of the transaction amount. If left blank, this value will default to the value specified in the Merchant Interface. See

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
				Appendix G for other values.
x_Method	Required	CC, ECHECK	N/A	Indicates the method of payment for the transaction being sent to the system. If left blank, this value will default to CC.
x_Type	Required	AUTH_CAPTURE, AUTH_ONLY, CAPTURE_ONLY, CREDIT, VOID, PRIOR_AUTH_CAPTURE	N/A	Indicates the type of transaction. If the value in the field does not match any of the values stated, the transaction will be rejected. If no value is submitted in this field, the gateway will process the transaction as an AUTH_CAPTURE
x_Recurring_Billing	Optional	YES, NO	3	Indicates whether the transaction is a recurring billing transaction.
x_Bank_ABA_Code	Conditional Required if x_Method = ECHECK	Valid routing number	9	Routing number of a bank for eCheck.Net transactions.
x_Bank_Acct_Num	Conditional Required if x_Method = ECHECK	Valid account number	20	Checking or savings account number.
x_Bank_Acct_Type	Conditional Required if x_Method = ECHECK	CHECKING, SAVINGS	8	Describes the type of bank account; if no value is provided, default is set to CHECKING.
x_Bank_Name	Conditional Required if x_Method = ECHECK	Valid bank name	50	Contains the name of the customer's financial institution.
x_Bank_Acct_Name	Conditional Required if x_Method = ECHECK	Name on the customer's bank account		Is the customer's name as it appears on their bank account.
x_Echeck_Type	Conditional Required if	WEB		This indicates that the eCheck payment request originated from a Website.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	x_Method = ECHECK			The system will default this value to WEB if no value is sent.
x_Card_Num	Conditional Required if x_Method = CC	Numeric credit card number	22	Contains the credit card number.
x_Exp_Date	Conditional Required if x_Method = CC	MMYY, MM/YY, MM-YY, MMYYYY, MM/YYYY, MM-YYYY, YYYY-MM-DD, YYYY/MM/DD	N/A	Contains the date on which the credit card expires.
x_Card_Code	Optional	Valid CVV2, CVC2 or CID value	4	Three- or four-digit number on the back of a credit card (on front for American Express).
x_Trans_ID	Conditional Required if x_Type = CREDIT, VOID, or PRIOR_AUTH_CAPTURE	Valid transaction ID	10	ID of a transaction previously authorized by the gateway.
x_Auth_Code	Conditional Required if x_Type = CAPTURE_ONLY	Valid authorization code	6	Authorization code for a previous transaction not authorized on the gateway that is being submitted for capture.

Level 2 Data

The system supports Level 2 transaction data by providing the following fields as part of the transaction submission API.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_PO_Num	Optional	Any string	25	Contains the purchase order number.
x_Tax	Optional	Any valid amount	15	Contains the tax amount.
x_Tax_Exempt	Optional	TRUE, FALSE	5	Indicates whether the transaction is tax exempt.
x_Freight	Optional	Any valid amount	10	Contains the freight amount charged.
x_Duty	Optional	Any valid amount	10	Contains the amount charged for duty.

Transaction Submission API for AIM Wells Fargo SecureSource Merchants

For merchants who process transactions through the Wells Fargo SecureSource product, some additional rules apply to transaction processing. Fields that are optional in the standard gateway API are required for Wells Fargo SecureSource merchants. The following tables describe these required fields. Only those fields that are different from the standard API are called out in this section.

Customer Name and Billing Address

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_First_Name	Required	Any string	50	Contains the first name of the customer associated with the billing address for the transaction.
x_Last_Name	Required	Any string	50	Contains the last name of the customer associated with the billing address for the transaction.
x_Company	Required	Any string	50	Contains the company name associated with the billing address for the transaction.
x_Address	Required	Any string	60	Contains the address of the customer associated with the billing address for the transaction.
x_City	Required	Any string	40	Contains the city of the customer associated with the billing address for the transaction.
x_State	Required	Any valid two-digit state code or full state name	40	Contains the state of the customer associated with the billing address for the transaction.
x_Zip	Required	Any string	20	Contains the zip of the customer associated with the billing address for the transaction.
x_Country	Required	Any valid two-digit country code or full country name (spelled in English)	60	Contains the country of the customer associated with the billing address for the transaction.
x_Phone	Required	Any string	25	Contains the phone number of the customer

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
		Recommended format is (123)123-1234		associated with the billing address for the transaction.

Email Settings

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Email	Required	Any valid email address	255	Email address to which a confirmation email is sent.

Additional Customer Data

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
x_Customer_IP	Required	Required format is 255.255.255.255. If this value is not passed, it will default to 255.255.255.255	15	IP address of the customer initiating the transaction.

x_Customer_Organization_Type	Required	I, B
		I = Individual
		B = Business
			N/A	Required for all eCheck transactions for Wells Fargo SecureSource Merchants.

x_Customer_Tax_ID	Conditional IF x_Type = ECHECK, merchant must provide EITHER x_Customer_Tax_ID OR x_Drivers_License_Num AND x_Drivers_License_State AND x_Drivers_License_DOB	9 digits or numbers only	9	Tax ID or SSN of the customer initiating the transaction. If the Tax ID or SSN is not available, the customer's driver's license number, driver's license state and date of birth must be used in its place.
x_Drivers_License_Num	Conditional IF x_Type = ECHECK, merchant must provide EITHER x_Customer_Tax_ID OR x Drivers License		50	Required for all eCheck transactions for Wells Fargo SecureSource Merchants where the Tax ID or SSN is not provided.

FIELD	REQUIRED	VALUE	MAX LENGTH	DESCRIPTION
	Num AND x_Drivers_License_ State AND x_Drivers_License_ DOB			
x_Drivers_License_State	Conditional IF x_Type = ECHECK, merchant must provide EITHER x_Customer_Tax_I D OR x_Drivers_License_ Num AND x_Drivers_License_ State AND x_Drivers_License_ DOB	2-character state abbreviation	2	Required for all eCheck transactions for Wells Fargo SecureSource Merchants where the Tax ID or SSN is not provided.
x_Drivers_License_DOB	Conditional IF x_Type = ECHECK, merchant must provide EITHER x_Customer_Tax_I D OR x_Drivers_License_ Num AND x_Drivers_License_ State AND x_Drivers_License_ DOB	YYYY-MM-DDD, YYYY/MM/DD, MM/DD/YYYY, MM-DD-YYYY,	N/A	Required for all eCheck transactions for Wells Fargo SecureSource Merchants where the Tax ID or SSN is not provided.

Gateway Response API

This section describes the response returned by the gateway when a merchant server submits a transaction for processing. The response is a set of fields that give merchants information about the status of a transaction. The fields will be comma delimited by default or delimited by the character specified by the merchant. The merchant server can parse this data and determine the message to display to the customer.

Fields in the Gateway Response

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

POSITION IN RESPONSE	FIELD NAME OF VALUE IN RESPONSE	DESCRIPTION
1	Response Code	Indicates the result of the transaction: 1 = Approved 2 = Declined 3 = Error
2	Response Subcode	A code used by the system for internal transaction tracking.
3	Response Reason Code	A code representing more details about the result of the transaction.
4	Response Reason Text	Brief description of the result, which corresponds with the Response Reason Code.
5	Approval Code	The six-digit alphanumeric authorization or approval code.

6	AVS Result Code	Indicates the result of Address Verification System (AVS) checks:
		A = Address (Street) matches, ZIP does not
		B = Address information not provided for AVS check
		E = AVS error
		G = Non-U.S. Card Issuing Bank
		N = No Match on Address (Street) or ZIP
		P = AVS not applicable for this transaction
		R = Retry – System unavailable or timed out
		S = Service not supported by issuer
		U = Address information is unavailable
		W = 9 digit ZIP matches, Address (Street) does not
		X = Address (Street) and 9 digit ZIP match
		Y = Address (Street) and 5 digit ZIP match
		Z = 5 digit ZIP matches, Address (Street) does not

7	Transaction ID	This number identifies the transaction in the system and can be used to submit a modification of this transaction at a later time, such as voiding, crediting or capturing the transaction.
8	Invoice Number	Echoed from form input value for x_Invoice_Num.
9	Description	Echoed from form input value for x_Description.
10	Amount	Echoed from form input value for x_Amount.
11	Method	Echoed from form input value for x_Method.
12	Transaction Type	Echoed from form input value for x_Type.
13	Customer ID	Echoed from form input value for x_Cust_ID.

POSITION IN RESPONSE	**FIELD NAME OF VALUE IN RESPONSE**	**DESCRIPTION**
14	Cardholder First Name	Echoed from form input value for x_First_Name.
15	Cardholder Last Name	Echoed from form input value for x_Last_Name.
16	Company	Echoed from form input value for x_Company.
17	Billing Address	Echoed from form input value for x_Address.
18	City	Echoed from form input value for x_City.
19	State	Echoed from form input value for x_State.
20	Zip	Echoed from form input value for x_Zip.
21	Country	Echoed from form input value for x_Country.
22	Phone	Echoed from form input value for x_Phone.
23	Fax	Echoed from form input value for x_Fax.
24	Email	Echoed from form input value for x_Email.
25	Ship to First Name	Echoed from form input value for x_Ship_To_First_Name.
26	Ship to Last Name	Echoed from form input value for x_Ship_To_Last_Name.
27	Ship to Company	Echoed from form input value for x_Ship_To_Company.
28	Ship to Address	Echoed from form input value for x_Ship_To_Address.
29	Ship to City	Echoed from form input value for x_Ship_To_City.
30	Ship to State	Echoed from form input value for x_Ship_To_State.
31	Ship to Zip	Echoed from form input value for x_Ship_To_Zip.
32	Ship to Country	Echoed from form input value for x_Ship_To_Country.
33	Tax Amount	Echoed from form input value for x_Tax.
34	Duty Amount	Echoed from form input value for x_Duty.
35	Freight Amount	Echoed from form input value for x_Freight.
36	Tax Exempt Flag	Echoed from form input value for x_Tax_Exempt.
37	PO Number	Echoed from form input value for x_PO_Num.
38	MD5 Hash	System-generated hash that may be validated by the merchant to authenticate a transaction response received from the gateway.

39	Card Code (CVV2/CVC2/CID) Response Code	Indicates the results of Card Code verification:
		M = Match
		N = No Match
		P = Not Processed
		S = Should have been present
		U = Issuer unable to process request

40 - 68		Reserved for future use.
69 -		Echo of merchant-defined fields.

AIM Transaction Response Types

There are two versions of the AIM response string:

Version 3.0

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

Version 3.1

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

Setting the Transaction Version

To set the transaction version, do the following:

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

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

Response Code Details

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

Description of Response Fields

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

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

Response Codes

RESPONSE CODE	DESCRIPTION
1	This transaction has been approved.
2	This transaction has been declined.
3	There has been an error processing this transaction.

Response Reason Codes & Response Reason Text

RESPONSE CODE	RESPONSE REASON CODE	RESPONSE REASON TEXT	NOTES
1	1	This transaction has been approved.	
2	2	This transaction has been declined.	
2	3	This transaction has been declined.	
2	4	This transaction has been declined.	The code returned from the processor indicating that the card used needs to be picked up.
3	5	A valid amount is required.	The value submitted in the amount field did not pass validation for a number.
3	6	The credit card number is invalid.	
3	7	The credit card expiration date is invalid.	The format of the date submitted was incorrect.
3	8	The credit card has expired.	
3	9	The ABA code is invalid.	The value submitted in the x_Bank_ABA_Code field did not pass validation or was not for a valid financial institution.
3	10	The account number is invalid.	The value submitted in the x_Bank_Acct_Num field did not pass validation.
3	11	A duplicate transaction has been submitted.	A transaction with identical amount and credit card information was submitted two minutes prior.
3	12	An authorization code is required but not present.	A transaction that required x_Auth_Code to be present was submitted without a value.
3	13	The merchant Login ID is invalid or the account is inactive.	
3	14	The Referrer or Relay Response URL is invalid.	The Relay Response or Referrer URL does not match the merchant's configured value(s) or is absent. Applicable only to SIM and WebLink APIs.
3	15	The transaction ID is invalid.	The transaction ID value is non-numeric or was not present for a transaction that requires it (i.e., VOID, PRIOR_AUTH_CAPTURE, and CREDIT).
3	16	The transaction was not found.	The transaction ID sent in was properly formatted but the gateway had no record of the transaction.
3	17	The merchant does not accept this type of credit card.	The merchant was not configured to accept the credit card submitted in the transaction.
3	18	ACH transactions are not accepted by this merchant.	The merchant does not accept electronic checks.
3	19	An error occurred during processing. Please try again in 5 minutes.	

3	20	An error occurred during processing. Please try again in 5 minutes.	
3	21	An error occurred during processing. Please try again in 5 minutes.	
3	22	An error occurred during processing. Please try again in 5 minutes.	
3	23	An error occurred during processing. Please try again in 5 minutes.	
3	24	The Nova Bank Number or Terminal ID is incorrect. Call Merchant Service Provider.	
3	25	An error occurred during processing. Please try again in 5 minutes.	
3	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.	
3	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.
3	29	The PaymentTech identification numbers are incorrect. Call Merchant Service Provider.	
3	30	The configuration with the processor is invalid. Call Merchant Service Provider.	
3	31	The FDC Merchant ID or Terminal ID is incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	32	This reason code is reserved or not applicable to this API.	
3	33	*FIELD* cannot be left blank.	The word *FIELD* will be replaced by an actual field name. This error indicates that a field the merchant specified as required was not filled in.
3	34	The VITAL identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	35	An error occurred during processing. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	36	The authorization was approved, but settlement failed.	
3	37	The credit card number is invalid.	

3	38	The Global Payment System identification numbers are incorrect. Call Merchant Service Provider.	The merchant was incorrectly set up at the processor.
3	39	The supplied currency code is either invalid, not supported, not allowed for this merchant or doesn’t have an exchange rate.	
3	40	This transaction must be encrypted.	
2	41	This transaction has been declined.	Only merchants set up for the FraudScreen.Net service would receive this decline. This code will be returned if a given transaction’s fraud score is higher than the threshold set by the merchant.
3	42	There is missing or invalid information in a required field.	This is applicable only to merchants processing through the Wells Fargo SecureSource product who have requirements for transaction submission that are different from merchants not processing through Wells Fargo.
3	43	The merchant was incorrectly set up at the processor. Call your Merchant Service Provider.	The merchant was incorrectly set up at the processor.
2	44	This transaction has been declined.	The merchant would receive this error if the Card Code filter has been set in the Merchant Interface and the transaction received an error code from the processor that matched the rejection criteria set by the merchant.
2	45	This transaction has been declined.	This error would be returned if the transaction received a code from the processor that matched the rejection criteria set by the merchant for both the AVS and Card Code filters.
3	46	Your session has expired or does not exist. You must log in to continue working.	
3	47	The amount requested for settlement may not be greater than the original amount authorized.	This occurs if the merchant tries to capture funds greater than the amount of the original authorization-only transaction.
3	48	This processor does not accept partial reversals.	The merchant attempted to settle for less than the originally authorized amount.
3	49	A transaction amount greater than $99,999 will not be accepted.	
3	50	This transaction is awaiting settlement and cannot be refunded.	Credits or refunds may only be performed against settled transactions. The transaction against which the credit/refund was submitted has not been settled, so a credit cannot be issued.
3	51	The sum of all credits against this transaction is greater than the original transaction amount.	
3	52	The transaction was authorized,	

		but the client could not be notified; the transaction will not be settled.	
3	53	The transaction type was invalid for ACH transactions.	If x_Method = ECHECK, x_Type cannot be set to CAPTURE_ONLY.
3	54	The referenced transaction does not meet the criteria for issuing a credit.	
3	55	The sum of credits against the referenced transaction would exceed the original debit amount.	The transaction is rejected if the sum of this credit and prior credits exceeds the original debit amount.
3	56	This merchant accepts ACH transactions only; no credit card transactions are accepted.	The merchant processes eCheck transactions only and does not accept credit cards.
3	57	An error occurred in processing. Please try again in 5 minutes.	
3	58	An error occurred in processing. Please try again in 5 minutes.	
3	59	An error occurred in processing. Please try again in 5 minutes.	
3	60	An error occurred in processing. Please try again in 5 minutes.	
3	61	An error occurred in processing. Please try again in 5 minutes.	
3	62	An error occurred in processing. Please try again in 5 minutes.	
3	63	An error occurred in processing. Please try again in 5 minutes.	
3	64	The referenced transaction was not approved.	This error is applicable to Wells Fargo SecureSource merchants only. Credits or refunds cannot be issued against transactions that were not authorized.
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	67	The given transaction type is not supported for this merchant.	This error code is applicable to merchants using the Wells Fargo SecureSource product only. This product does not allow transactions of type CAPTURE_ONLY.
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 six characters in length.

3	73	The driver's license date of birth is invalid.	The format of the value submitted in x_Drivers_License_Num was invalid.
3	74	The duty amount is invalid.	The value submitted in x_Duty failed format validation.
3	75	The freight amount is invalid.	The value submitted in x_Freight failed format validation.
3	76	The tax amount is invalid.	The value submitted in x_Tax failed format validation.
3	77	The SSN or tax ID is invalid.	The value submitted in x_Customer_Tax_ID failed validation.
3	78	The Card Code (CVV2/CVC2/CID) is invalid.	The value submitted in x_Card_Code failed format validation.
3	79	The driver's license number is invalid.	The value submitted in x_Drivers_License_Num failed format validation.
3	80	The driver's license state is invalid.	The value submitted in x_Drivers_License_State failed format validation.
3	81	The requested form type is invalid.	The merchant requested an integration method not compatible with the AIM API.
3	82	Scripts are only supported in version 2.5.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	83	The requested script is either invalid or no longer supported.	The system no longer supports version 2.5; requests cannot be posted to scripts.
3	84	This reason code is reserved or not applicable to this API.	
3	85	This reason code is reserved or not applicable to this API.	
3	86	This reason code is reserved or not applicable to this API.	
3	87	This reason code is reserved or not applicable to this API.	
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	93	A valid country is required.	This code is applicable to Wells Fargo SecureSource merchants only. Country is a required field and must contain the value of a supported country.
3	94	The shipping state or country is invalid.	This code is applicable to Wells Fargo SecureSource merchants only.
3	95	A valid state is required.	This code is applicable to Wells Fargo SecureSource merchants only.

3	96	This country is not authorized for buyers.	This code is applicable to Wells Fargo SecureSource merchants only. Country is a required field and must contain the value of a supported country.
3	97	This transaction cannot be accepted.	Applicable only to SIM API. Fingerprints are only valid for a short period of time. This code indicates that the transaction fingerprint has expired.
3	98	This transaction cannot be accepted.	Applicable only to SIM API. The transaction fingerprint has already been used.
3	99	This transaction cannot be accepted.	Applicable only to SIM API. The server-generated fingerprint does not match the merchant-specified fingerprint in the x_FP_Hash field.
3	100	The eCheck type is invalid.	Applicable only to eCheck. The value specified in the x_Echeck_type field is invalid.
3	101	The given name on the account and/or the account type does not match the actual account.	Applicable only to eCheck. The specified name on the account and/or the account type do not match the NOC record for this account.
3	102	This request cannot be accepted.	A password or transaction key was submitted with this WebLink request. This is a high security risk.
3	103	This transaction cannot be accepted.	A valid fingerprint, transaction key, or password is required for this transaction.
3	104	This transaction is currently under review.	Applicable only to eCheck. The value submitted for country failed validation.
3	105	This transaction is currently under review.	Applicable only to eCheck. The values submitted for city and country failed validation.
3	106	This transaction is currently under review.	Applicable only to eCheck. The value submitted for company failed validation.
3	107	This transaction is currently under review.	Applicable only to eCheck. The value submitted for bank account name failed validation.
3	108	This transaction is currently under review.	Applicable only to eCheck. The values submitted for first name and last name failed validation.
3	109	This transaction is currently under review.	Applicable only to eCheck. The values submitted for first name and last name failed validation.
3	110	This transaction is currently under review.	Applicable only to eCheck. The value submitted for bank account name does not contain valid characters.
3	111	A valid billing country is required.	This code is applicable to Wells Fargo SecureSource merchants only.
3	112	A valid billing state/provice is required.	This code is applicable to Wells Fargo SecureSource merchants only.
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.
2	141	This transaction has been	The system-generated void for the original

		declined.	FraudScreen-rejected transaction failed.
2	145	This transaction has been declined.	The system-generated void for the original card code-rejected and AVS-rejected transaction failed.
2	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.

Note: Response code reasons that are not included in numerical order are reserved, or may not be applicable to this API.

Appendix A – Types of Credit Card Transactions

There are two steps to credit card transaction processing:

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

   Note: The merchant can specify when the last transaction is picked up for settlement by the gateway. To modify the Transaction Cut-Off Time, do the following:
   1. Log into the Merchant Interface
   2. Select *Settings*
   3. Select *Transaction Cut-Off Time* from the General section
   4. Using the drop-down boxes, select the desired cut-off time
   5. Click *Submit* to save changes

Credit Card Transaction Types

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

TRANSACTION TYPE	**DESCRIPTION**
AUTH_CAPTURE	Transactions of this type will be sent for authorization. The transaction will be automatically picked up for settlement if approved. This is the default transaction type in the gateway. If no type is indicated when submitting transactions to the gateway, the gateway will assume that the transaction is of the type AUTH_CAPTURE.
AUTH_ONLY	Transactions of this type are submitted if the merchant wishes to validate the credit card for the amount of the goods sold. If the merchant does not have goods in stock or wishes to review orders before shipping the goods, this transaction type should be submitted. The gateway will send this type of transaction to the financial institution for approval. However this transaction will not be sent for settlement. If the merchant does not act on the transaction within 30 days, the transaction will no longer be available for capture.
PRIOR_AUTH_CAPTURE	This transaction is used to request settlement for a transaction that was previously submitted as an AUTH_ONLY. The gateway will accept this transaction and initiate settlement if the following conditions are met: • The transaction is submitted with the ID of the original authorization-only transaction, which needs to be settled. • The transaction ID is valid and the system has a record of the original authorization-only transaction being submitted. • The original transaction referred to is not already settled or expired or errored.

	• The amount being requested for settlement in this transaction is less than or equal to the original authorized amount. If no amount is submitted in this transaction, the gateway will initiate settlement for the amount of the originally authorized transaction. In addition to the required fields in the API, the following is required to submit a PRIOR_AUTH_CAPTURE type transaction: • x_Version = 3.1 • x_Login = merchant Login ID • x_Tran_Key = transaction key • x_Trans_ID = the transaction ID of the previously authorized transaction
CREDIT	This transaction is also referred to as a "Refund" and indicates to the gateway that money should flow from the merchant to the customer. The gateway will accept a credit or a refund request if the transaction submitted meets the following conditions: • The transaction is submitted with the ID of the original transaction against which the credit is being issued (x_Trans_ID). • The gateway has a record of the original transaction. • The original transaction has been settled. • The sum of the amount submitted in the Credit transaction and all credits submitted against the original transaction is less than the original transaction amount. • The first and last four digits of the credit card number submitted with the credit transaction match the first and last four digits of the credit card number used in the original transaction. If no credit card number is submitted with the transaction and all the checks pass, the Credit will be issued against the credit card used in the original transaction. A transaction key is required to submit a credit to the system (i.e., x_Tran_Key should have a valid value when a CREDIT transaction is submitted).
CAPTURE_ONLY	This is a request to settle a transaction that was not submitted for authorization through the payment gateway. The gateway will accept this transaction if an authorization code is submitted. x_Auth_Code is a required field for CAPTURE_ONLY type transactions.
VOID	This transaction is an action on a previous transaction and is used to cancel the previous transaction and ensure it does not get sent for settlement. It can be done on any type of transaction (i.e., CREDIT, AUTH_CAPTURE, CAPTURE_ONLY, and AUTH_ONLY). The transaction will be accepted by the gateway if the following conditions are met: • The transaction is submitted with the ID of the transaction that has to be voided. • The gateway has a record of the transaction referenced by the ID. • The transaction has not been sent for settlement. For a transaction of type VOID, the following fields are required (in addition to the other required fields in the API): • x_Version = 3.1 • x_Login = merchant Login ID • x_Tran_Key = merchant transaction key • x_Trans_ID = the transaction ID that needs to be voided

Appendix B – Features of the Gateway

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

Address Verification System

The Address Verification System (AVS) helps merchants to detect suspicious transaction activity. To use this system, the merchant must submit the customer's credit card billing address to the gateway for validation. This information is submitted by the gateway to the financial institutions. The financial institutions compare the submitted address with the billing address on file for that particular credit card and return an AVS response code to the gateway. The gateway includes this code in the response back to the merchant.

The merchant can configure the gateway to reject or accept transactions based on the AVS code returned. To configure rejection or acceptance of a transaction based on the AVS code, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Address Verification System (AVS)* link from the Security section
4. Check the box(es) next to the AVS codes that the system should reject
5. Click *Submit* to save changes

AVS CODE	DESCRIPTION (Italics denote a default setting)
A	Address (Street) matches, ZIP does not
B	*Address information not provided for AVS check*
E	*AVS error*
G	*Non-U.S. Card Issuing Bank*
N	*No Match on Address (Street) or ZIP*
P	AVS not applicable for this transaction
R	*Retry – System unavailable or timed out*
S	*Service not supported by issuer*
U	*Address information is unavailable*
W	9 digit ZIP matches, Address (Street) does not
X	Address (Street) and 9 digit ZIP match
Y	Address (Street) and 5 digit ZIP match
Z	5 digit ZIP matches, Address (Street) does not

Note: It is recommended that merchants enable some level of Address Verification to avoid non-qualified transaction surcharges that can be levied by merchant banks and merchant service providers. Please note, however, that the merchant will incur applicable transaction fees for transactions that are declined due to an AVS mismatch (as with any other declined transaction). System defaults are marked in italics in the table above.

Credit Card Identification Code (CVV2/CVC2/CID)

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

To configure the filter to reject certain Card Code responses, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Card Code Verification* link from the Security section
4. Check the box(es) next to the Card Codes that the system should reject
5. Click *Submit* to save changes

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

Appendix C – Customizing Notification to Customers

Merchants will be sent a confirmation email after the gateway completes processing on a transaction submitted to the system. The confirmation email enables merchants to know the results of a given transaction. Multiple contacts can be configured to receive these email notifications. Additionally, merchants can choose to send a confirmation email to their customers.

Configuration of these contacts can be done through the Merchant Interface:

1. Log into the Merchant Interface
2. Click on the *Settings* link from the left navigation bar
3. Click on the *Email Receipts* link from the Transaction Response section
4. Check the box if email receipts should be sent to the customer
5. Configure the header and footer of the email message
6. Click *Submit* to save changes

It is possible to configure the confirmation email on a per-transaction basis by submitting the information with each transaction. The following table describes the fields used in the API to configure email notification to the customer; all fields are optional.

FIELD	VALUE	DESCRIPTION
x_Email_Customer	TRUE, FALSE	If set to TRUE, the gateway will send an email to the customer after the transaction is processed using the customer email supplied in the transaction. If FALSE, no email will be sent to the customer. If no value is submitted, the gateway will look up the configuration in the Merchant Interface and send an email only if the merchant has configured the option to be TRUE. If there are no incoming parameters and the merchant has not configured this option, no email will be sent to the customer.
x_Header_Email_Receipt	Any valid text	This text will appear as the header on the transaction confirmation email sent to the customer.
x_Footer_Email_Receipt	Any valid text	This text will appear as the footer on the transaction confirmation email sent to the customer.

Appendix D – The MD5 Hash Security Feature

What is the MD5 Hash Security Feature?

The MD5 Hash security feature enables merchants to verify that the results of a transaction received by their server were actually sent from the Payment Gateway. The MD5 Hash works likes this:

1. The merchant sets a value in the Merchant Interface
2. The gateway uses this value, along with a predefined set of fields submitted in the transaction, to create a unique signature
3. The merchant server that receives the transaction response containing this signature determines whether it was returned from the gateway

The mathematical algorithm used to construct this signature is designed in such a way that any change to the information used in its calculation will cause a completely different signature to be created. Also, the information used in the calculation of the signature cannot be discovered through any analysis of the signature itself.

How is the Signature Constructed?

The MD5 signature is a hash of the following four fields: MD5 Hash Value, Login ID, Transaction ID, and Amount, in the following order:

"MD5 Hash Value" "Login ID" "Trans ID" "Amount"

For example, if the merchant's hash value was "wilson," the merchant Login ID was "mylogin," the transaction ID was "987654321," and the amount was "1.00," the MD5 algorithm would be run on the following string:

"wilsonmylogin9876543211.00"

Note: The value passed in *x_Amount* is formatted with the correct number of decimal places and the decimal point for the type of currency used in the transaction. For transactions that do not include a transacation amount, mainly VOIDs, the amount used to calculate the MD5 Hash is formatted as 0.00.

How Should the Feature be Set Up on the Merchant's Server?

The following steps are used by the merchant to evaluate the MD5 signature:

1. Create a script to receive transaction results
2. Run the MD5 algorithm on the fields indicated above
3. Determine if the signature created matches the signature that was returned by the gateway
4. If the signatures match, the response was sent by the gateway

How is the MD5 Hash Value Set Up in the Merchant Interface?

To set the MD5 Hash Value in the Merchant Interface, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on *MD5 Hash* in the Security section
4. Enter the MD5 Hash Value
5. Confirm the MD5 Hash Value entered
6. Click *Submit* to save changes

Appendix E – Submitting Test Transactions to the System

Test Mode

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

To set an account to Test Mode, do the following:

1. Log into the Merchant Interface
2. Select *Settings* from the Main Menu
3. Click on the *Test Mode* Link in the General section
4. Click on the *Turn Test On* button

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

Note: Test Mode is only supported if the merchant is submitting transactions from a Website or through the Virtual Terminal. If the merchant uploads a file of transactions for offline processing, the gateway will reject the file.

Running a Test Transaction

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

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

VALUE PASSED IN X_TEST_REQUEST	CONFIGURATION IN MERCHANT INTERFACE	GATEWAY BEHAVIOR
TRUE	ON	Transaction processed as test
FALSE	ON	Transaction processed as test
TRUE	OFF	Transaction processed as test
FALSE	OFF	Transaction processed as a live transaction

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

Test credit card numbers

Any of the following card numbers can be used to run test transactions. Please note that these numbers do not represent real card accounts; they will return a decline in live mode, and an approval in test mode. Any expiration dates after the current day's date can be used with these numbers.

TEST CARD NUMBER	CARD TYPE
370000000000002	American Express
6011000000000012	Discover
5424000000000015	MasterCard
4007000000027	Visa

There is also a test credit card number that can be used to generate errors. THIS CARD IS INTENDED TO PRODUCE ERRORS, and should only be used if that is the intent.

To cause the system to generate a specific error, set the account to Test Mode and submit a transaction with the card number 4222222222222. The system will return the response reason code equal to the amount of the submitted transaction. For example, to test response reason code number 27, a test transaction would be submitted with the credit card number, “4222222222222,” and the amount, “27.00.”

Appendix F – Certification

It is possible for a merchant to test their integration using a test gateway system. In order to test the integration, the merchant should post transactions to **https://certification.authorize.net/gateway/transact.dll**. The test gateway behavior will be identical to the primary gateway. Transactions sent to the test gateway are not submitted to financial institutions for authorization, will not be stored on the system and cannot be retrieved from the system (as is the case when using Test Mode set to TRUE with the primary gateway system).

Appendix G – Currency Codes

CURRENCY COUNTRY	CURRENCY CODE
Afghani (Afghanistan)	AFA
Algerian Dinar (Algeria)	DZD
Andorran Peseta (Andorra)	ADP
Argentine Peso (Argentina)	ARS
Armenian Dram (Armenia)	AMD
Aruban Guilder (Aruba)	AWG
Australian Dollar (Australia)	AUD
Azerbaijanian Manat (Azerbaijan)	AZM
Bahamian Dollar (Bahamas)	BSD
Bahraini Dinar (Bahrain)	BHD
Baht (Thailand)	THB
Balboa (Panama)	PAB
Barbados Dollar (Barbados)	BBD
Belarussian Ruble (Belarus)	BYB
Belgian Franc (Belgium)	BEF
Belize Dollar (Belize)	BZD
Bermudian Dollar (Bermuda)	BMD
Bolivar (Venezuela)	VEB
Boliviano (Bolivia)	BOB
Brazilian Real (Brazil)	BRL
Brunei Dollar (Brunei Darussalam)	BND
Bulgarian Lev (Bulgaria)	BGN
Burundi Franc (Burundi)	BIF
Canadian Dollar (Canada)	CAD
Cape Verde Escudo (Cape Verde)	CVE
Cayman Islands Dollar (Cayman Islands)	KYD
Cedi (Ghana)	GHC
CFA Franc BCEAO (Guinea-Bissau)	XOF
CFA Franc BEAC (Central African Republic)	XAF
CFP Franc (New Caledonia)	XPF
Chilean Peso (Chile)	CLP
Colombian Peso (Colombia)	COP
Comoro Franc (Comoros)	KMF
Convertible Marks (Bosnia And Herzegovina)	BAM
Cordoba Oro (Nicaragua)	NIO
Costa Rican Colon (Costa Rica)	CRC
Cuban Peso (Cuba)	CUP
Cyprus Pound (Cyprus)	CYP
Czech Koruna (Czech Republic)	CZK
Dalasi (Gambia)	GMD
Danish Krone (Denmark)	DKK
Denar (The Former Yugoslav Republic Of Macedonia)	MKD
Deutsche Mark (Germany)	DEM
Dirham (United Arab Emirates)	AED
Djibouti Franc (Djibouti)	DJF
Dobra (Sao Tome And Principe)	STD

Dominican Peso (Dominican Republic)	DOP
Dong (Vietnam)	VND
Drachma (Greece)	GRD
East Caribbean Dollar (Grenada)	XCD
Egyptian Pound (Egypt)	EGP
El Salvador Colon (El Salvador)	SVC
Ethiopian Birr (Ethiopia)	ETB
Euro (Europe)	EUR
Falkland Islands Pound (Falkland Islands)	FKP
Fiji Dollar (Fiji)	FJD
Forint (Hungary)	HUF
Franc Congolais (The Democratic Republic Of Congo)	CDF
French Franc (France)	FRF
Gibraltar Pound (Gibraltar)	GIP
Gold	XAU
Gourde (Haiti)	HTG
Guarani (Paraguay)	PYG
Guinea Franc (Guinea)	GNF
Guinea-Bissau Peso (Guinea-Bissau)	GWP
Guyana Dollar (Guyana)	GYD
Hong Kong Dollar (Hong Kong)	HKD
Hryvnia (Ukraine)	UAH
Iceland Krona (Iceland)	ISK
Indian Rupee (India)	INR
Iranian Rial (Islamic Republic Of Iran)	IRR
Iraqi Dinar (Iraq)	IQD
Irish Pound (Ireland)	IEP
Italian Lira (Italy)	ITL
Jamaican Dollar (Jamaica)	JMD
Jordanian Dinar (Jordan)	JOD
Kenyan Shilling (Kenya)	KES
Kina (Papua New Guinea)	PGK
Kip (Lao People's Democratic Republic)	LAK
Kroon (Estonia)	EEK
Kuna (Croatia)	HRK
Kuwaiti Dinar (Kuwait)	KWD
Kwacha (Malawi)	MWK
Kwacha (Zambia)	ZMK
Kwanza Reajustado (Angola)	AOR
Kyat (Myanmar)	MMK
Lari (Georgia)	GEL
Latvian Lats (Latvia)	LVL
Lebanese Pound (Lebanon)	LBP
Lek (Albania)	ALL
Lempira (Honduras)	HNL
Leone (Sierra Leone)	SLL
Leu (Romania)	ROL
Lev (Bulgaria)	BGL
Liberian Dollar (Liberia)	LRD
Libyan Dinar (Libyan Arab Jamahiriya)	LYD

Lilangeni (Swaziland)	SZL
Lithuanian Litas (Lithuania)	LTL
Loti (Lesotho)	LSL
Luxembourg Franc (Luxembourg)	LUF
Malagasy Franc (Madagascar)	MGF
Malaysian Ringgit (Malaysia)	MYR
Maltese Lira (Malta)	MTL
Manat (Turkmenistan)	TMM
Markka (Finland)	FIM
Mauritius Rupee (Mauritius)	MUR
Metical (Mozambique)	MZM
Mexican Peso (Mexico)	MXN
Mexican Unidad de Inversion (Mexico)	MXV
Moldovan Leu (Republic Of Moldova)	MDL
Moroccan Dirham (Morocco)	MAD
Mvdol (Bolivia)	BOV
Naira (Nigeria)	NGN
Nakfa (Eritrea)	ERN
Namibia Dollar (Namibia)	NAD
Nepalese Rupee (Nepal)	NPR
Netherlands (Netherlands)	ANG
Netherlands Guilder (Netherlands)	NLG
New Dinar (Yugoslavia)	YUM
New Israeli Sheqel (Israel)	ILS
New Kwanza (Angola)	AON
New Taiwan Dollar (Province Of China Taiwan)	TWD
New Zaire (Zaire)	ZRN
New Zealand Dollar (New Zealand)	NZD
Ngultrum (Bhutan)	BTN
North Korean Won (Democratic People's Republic Of Korea)	KPW
Norwegian Krone (Norway)	NOK
Nuevo Sol (Peru)	PEN
Ouguiya (Mauritania)	MRO
Pa'anga (Tonga)	TOP
Pakistan Rupee (Pakistan)	PKR
Palladium	XPD
Pataca (Macau)	MOP
Peso Uruguayo (Uruguay)	UYU
Philippine Peso (Philippines)	PHP
Platinum	XPT
Portuguese Escudo (Portugal)	PTE
Pound Sterling (United Kingdom)	GBP
Pula (Botswana)	BWP
Qatari Rial (Qatar)	QAR
Quetzal (Guatemala)	GTQ
Rand (Financial) (Lesotho)	ZAL
Rand (South Africa)	ZAR
Rial Omani (Oman)	OMR
Riel (Cambodia)	KHR
Rufiyaa (Maldives)	MVR

Rupiah (Indonesia)	IDR
Russian Ruble (Russian Federation)	RUB
Russian Ruble (Russian Federation)	RUR
Rwanda Franc (Rwanda)	RWF
Saudi Riyal (Saudi Arabia)	SAR
Schilling (Austria)	ATS
Seychelles Rupee (Seychelles)	SCR
Silver	XAG
Singapore Dollar (Singapore)	SGD
Slovak Koruna (Slovakia)	SKK
Solomon Islands Dollar (Solomon Islands)	SBD
Som (Kyrgyzstan)	KGS
Somali Shilling (Somalia)	SOS
Spanish Peseta (Spain)	ESP
Sri Lanka Rupee (Sri Lanka)	LKR
St Helena Pound (St Helena)	SHP
Sucre (Ecuador)	ECS
Sudanese Dinar (Sudan)	SDD
Surinam Guilder (Suriname)	SRG
Swedish Krona (Sweden)	SEK
Swiss Franc (Switzerland)	CHF
Syrian Pound (Syrian Arab Republic)	SYP
Tajik Ruble (Tajikistan)	TJR
Taka (Bangladesh)	BDT
Tala (Samoa)	WST
Tanzanian Shilling (United Republic Of Tanzania)	TZS
Tenge (Kazakhstan)	KZT
Timor Escudo (East Timor)	TPE
Tolar (Slovenia)	SIT
Trinidad and Tobago Dollar (Trinidad And Tobago)	TTD
Tugrik (Mongolia)	MNT
Tunisian Dinar (Tunisia)	TND
Turkish Lira (Turkey)	TRL
Uganda Shilling (Uganda)	UGX
Unidad de Valor Constante (Ecuador)	ECV
Unidades de fomento (Chile)	CLF
US Dollar (Next day) (United States)	USN
US Dollar (Same day) (United States)	USS
US Dollar (United States)	USD
Uzbekistan Sum (Uzbekistan)	UZS
Vatu (Vanuatu)	VUV
Won (Republic Of Korea)	KRW
Yemeni Rial (Yemen)	YER
Yen (Japan)	JPY
Yuan Renminbi (China)	CNY
Zimbabwe Dollar (Zimbabwe)	ZWD
Zloty (Poland)	PLN