



# **SecureNet Gateway API Implementation Guide**

## **Card-Not-Present Transactions and Card Present Transactions**

**Version 4.0**

**CONFIDENTIAL**

Last Revised June 2010

All rights reserved. SecureNet Payment Systems, [www.securenet.com](http://www.securenet.com)

# Contents

<b>Introduction .....</b>	<b>1</b>
<b>Gateway.....</b>	<b>2</b>
What is the Gateway? .....	2
Service Based Authorization Process .....	2
How Does the Gateway Work? .....	2
What is Required to Implement the Gateway? .....	2
<b>Gateway Implementation.....</b>	<b>3</b>
Service and SOAP Overview.....	3
Authorization Components .....	4
Request.....	4
Response .....	4
Gateway Service Request Definition.....	4
<b>Transaction Request Objects .....</b>	<b>5</b>
Standard Payment Transactions .....	5
SecureNet Vault/Tokenization Transactions .....	5
Transaction Response Object .....	5
Transaction Submission .....	6
TRANSACTION.....	7
AUTO.....	10
CARD.....	10
CHECK .....	10
CUSTOMER_BILL.....	11
CUSTOMER_SHIP.....	11
HOTEL.....	11
LEVEL2 .....	12
LEVEL3 .....	12
LEVEL3PRODUCT.....	13
MERCHANT_KEY .....	13
MPI .....	13
PETROLEUM .....	14
PRODUCT .....	14
SERVICE.....	15
USERDEFINED.....	15
<b>VAULT OBJECTS .....</b>	<b>15</b>
TRANSACTION_VAULT.....	15
ACCOUNT_VAULT.....	16
CUSTOMER_VAULT .....	16
OPERATIONPARAMETERS .....	17
TRANSACTIONIMAGE .....	17
CHECKIMAGE .....	17
<b>GATEWAYRESPONSE.....</b>	<b>17</b>
GATEWAYRESPONSE .....	17
TRANSACTIONRESPONSE.....	18
VAULTACCOUNTRESPONSE .....	20
VAULTCUSTOMERRESPONSE.....	20
<b>Method Overview .....</b>	<b>21</b>
<b>Appendix .....</b>	<b>24</b>
<b>Test Data .....</b>	<b>34</b>
<b>Sample Code (C#) .....</b>	<b>35</b>

# 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. The Gateway acts as a bridge between the merchant's website/terminal and the financial institutions that process payment transactions. Payment data is collected 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. 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.

The Gateway is the recommended integration method for merchants who have the capability to initiate 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, 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.

# Gateway

## What is the Gateway?

The Gateway is the recommended method of submitting transaction. 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.

## Service Based Authorization Process

Payment authorization utilizes a standards-based Simple Object Access Protocol (SOAP) interface over a Hyper Text Transfer Protocol (HTTP) connection to exchange information.

## How Does the Gateway Work?

When using the Gateway, transactions flow in the following way:

1. The customer's program has a reference to the Gateway Service and connects securely to the merchant's server to transmit payment information.
2. The merchant's server initiates a secure connection to the Gateway and creates the objects as described. It then initiates a "ProcessTransaction" function by passing the TRANSACTION object along with required data to the gateway server.
3. The payment gateway receives and processes the TRANSACTION object.
4. The payment gateway generates the GATEWAYRESPONSE object and submits it to the merchant's server.
5. The merchant's server receives and processes the GATEWAYRESPONSE object.
6. Finally, the merchant's server communicates the success (or failure) of the authorization to the customer's web browser or program.

## What is Required to Implement the Gateway?

Merchants must be able to perform the following functions in order to submit transactions to the Gateway:

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

# Gateway Implementation

## Service and SOAP Overview

The Gateway has implemented WCF (Windows Communication Foundation) as an underlying technology to support server-to-server authorization capabilities. WCF Services are applications with logic and functions, which are accessible and reusable by means of standard Internet protocols and XML data formats.

The WCF service interface is defined in terms of the XML messages that the service accepts and generates. Applications consuming WCF Services can be implemented on any platform in any programming language, as long as they can create and consume messages defined for the service interface. The Gateway has adopted Simple Object Access Protocol (SOAP) as the solution for initiating RPC-based authorization requests. SOAP is a lightweight protocol intended for exchanging structured information in a distributed environment. The protocol uses XML technologies to define an extensible messaging framework to provide a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model or language, platform, and other implementation-specific semantics.

To implement the Gateway service, a developer would design a script that performs the following:

1. Securely obtain all the information needed to process a transaction.
2. Create a program to consume the Gateway service.  
URL for Live merchant accounts: <https://gateway.securenet.com/api/Gateway.svc>  
URL for Test merchant accounts: <https://certify.securenet.com/api/Gateway.svc>

**Note:** The SecureNet Gateway will only accept transactions on port 443. This request object will include all system variables mentioned in the tables below.

3. Receive and process the response from the Gateway to display the appropriate result to the end-user.

## Authorization Components

*An authorization consists of the following:*

### Request

Authorization Request Objects sent to Gateway for processing. One authorization can be sent per request to Gateway. Batch authorizations are not supported.

### Response

Authorization Response Objects sent to the Merchant as a response to each Authorization Request.

## Gateway Service Request Definition

The following is the method signature for the ProcessTransaction service

```
[ServiceNamespace].GatewayClient Client = new [ServiceNamespace].GatewayClient("[ServiceBinding]")
oG = Client.ProcessTransaction(oT);
Client.Close();
```

Client: WCF service object

oT: TRANSACTION object

oG: GATEWAYRESPONSE object

ProcessTransaction is initiating Authorization Process.

## Transaction Request Objects

The objects submitted to Gateway to request an authorization are shown below.  
Parameters for the objects are described in this document.

### Standard Payment Transactions

Object Name	Description	Restrictions
TRANSACTION	The parameters defined in the tables below are assigned to the instance of this serialized object and passed to the web method "ProcessTransaction" for completing an authorization.	All restrictions as defined in the Request Object Specs apply

### SecureNet Vault/Tokenization Transactions

Object Name	Description	Restrictions
ACCOUNT_VAULT	Contains account information for a specific customer within the SecureNet Vault	Must be enabled to use the SecureNet Vault
CUSTOMER_VAULT	Contains customer information for a specific customer within the SecureNet Vault	Must be enabled to use the SecureNet Vault
TRANSACTION_VAULT	Used to process a SecureNet Vault transaction	Must be enabled to use the SecureNet Vault

### Transaction Response Object

Object Name	Description	Restrictions
GATEWAYRESPONSE	Response object returned by the gateway with all the values defined below.	

## Transaction Submission

The following tables define 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 Value set to a corresponding property of an Object. The Property is the field name and indicates to the gateway what information is being submitted. The Value contains the content of the field.

The following tables contain the data fields that may be submitted to the gateway with any transaction. The fields are grouped alphabetically in the tables.

Each table represents an object and contains the following information:

- **Field** – Name of the parameter that may be submitted in a transaction.
- **Required** – Indicates whether the field is required in a transaction. 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.
- **Type** – Property Type.
- **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.



## TRANSACTION

Property Name	Required	Type	Value	Max Length	Description
AMOUNT	True	Decimal	Any Amount > 0 and < 100,000	15	Total value to be charged or credited. Include decimal point followed by decimal amount
AUTHCODE	False	String	Valid authorization code	6	Authorization code for Offline Transaction 0300 (Capture Only). Voice Authorization Code.
CASHBACK_AMOUNT	False	Decimal	Any Amount >= 0 and <= 100	15	Cash back amount for card transactions only Include decimal point followed by decimal amount
CODE	False	String		4	Transaction Codes- indicates the type of processing required for the transaction request <a href="#">See Appendix</a>
CUSTOMERID <sup>†</sup>	False	String	Any String	25	Contains the customer ID associated with the transaction.
CUSTOMERIP	False	String		15	Customer IP address Include Customer's IP that's needs to be associated with the transaction. Recommended for eCommerce Merchant
DCI <sup>†</sup>	True	Integer	0, 1, 2, 3	1	Duplicate Check Indicator <a href="#">See Appendix</a>
DEVICECODE	False	String	Blank unless specified/ provided from SecureNet	2	Code for payment device
ENTRYSOURCE	False	String	Blank unless specified/ provided from SecureNet	2	Transaction Source
INDUSTRYSPECIFICDATA	False	String	P, D, 1, 2, 3	1	Industry Specific Data – Required to send the right indicator for transaction <a href="#">See Appendix</a>
INSTALLMENT_SEQUENCENUM	False <sup>†††</sup>	Integer	1 - 99	2	Installment number for installment payment plan

Property Name	Required	Type	Value	Max Length	Description
INVOICEDESC	False	String	Any String	50	Description of the transaction for which invoice is being issued.
INVOICENUM	False	String	Any String	50	Invoice number associated with the transaction.
MARKETSPECIFICDATA <sup>†</sup>	False	String	B	2	For bill payment transaction only. <a href="#">See Appendix</a>
METHOD	False	String	CC, ECHECK, CHECK21, DB, SV, PD	7	Transaction Method <a href="#">See Appendix</a>
NOTE	False	String		500	Additional Transaction Note
ORDERID	True	String A-Z, 0-9 only		25	Transaction Order ID <a href="#">See Appendix</a>
OVERRIDE_FROM	False	String	0, 1, 2	1	0 – Transaction 1 – Customer 2 – Account
PAYMENTID <sup>†</sup>	False	String		25	SecureNet Vault Payment ID
REF_TRANSID	False	String		15	Reference Transaction ID <a href="#">See Appendix</a>
RETAIL_LANENUM	False	String		6	Lane Number for Retail stores
SOFTDESCRIPTOR	False	String	Any String	25	Additional DBA descriptor <a href="#">See Appendix</a>
TEST	False	String	TRUE / FALSE	5	TRUE = Test Transaction FALSE = Live Transaction
TOTAL_INSTALLMENTCOUNT	False <sup>†††</sup>	Integer	1-99	2	Number of total installments for installment plans
TRANSACTION_SERVICE	True	Integer	0,1,2,3	1	Indicates Transaction Handling when Vault is enabled <a href="#">See Appendix</a>
AUTO <sup>††</sup>	False	Object			Object used for Rental industry
CARD	False *	Object			Card Payment Object
CHECK	False **	Object			Check Payment Object
CUSTOMER_BILL	False	Object			Object Containing the billing customer's information

Property Name	Required	Type	Value	Max Length	Description
CUSTOMER_SHIP	False	Object			Object Containing the shipping customer's information
HOTEL <sup>††</sup>	False	Object			Object used for Hotel industry
LEVEL2	False	Object			Object used for Level 2 data
LEVEL3	False	Object			Object used for Level 3 data
MERCHANT_KEY	True	Object			Object used to verify merchant before processing any transaction
MPI	False	Object			Merchant Plug-in Indicator for 3D Secure Transactions
PETROLEUM <sup>††</sup>	False	Object			Object used for Petroleum industry
PRODUCTS	False	Object ARRAY			Array of PRODUCT objects containing individual product information
SECONDARY_MERCHANT_KEY	False	Object			Object used to verify secondary merchant
SERVICE <sup>††</sup>	False	Object			Object used for Restaurant industry
USERDEFINED	False	Object			Object containing 50 user defined fields

<sup>†</sup> Must be enabled on your account

<sup>††</sup> Specific to Industry

<sup>†††</sup> Required if INDUSTRYSPECIFICDATA = 3

\* Require if METHOD = CC, PD, DB, or SV

\*\* Required if METHOD = ECHECK or CHECK21

**AUTO:**

Property Name	Required	Type	Value	Max Length	Description
EXTRA_CHARGEAMOUNT	False	Decimal		15	Additional Charge Amount
EXTRA_CHARGECODE	False	String		10	Additional Charge Code
PREFERRED_CUSTOMER	False	String	Y/N	1	Preferred Customer
RENTALCITY	False	String		50	City Rented in
RENTALDATETIME	False	String		14	mmddyyyyhhMMss
RENTALSTATE	False	String		2	State Rented in
RETURNCITY	False	String		59	City Returned In
RETURNDATETIME	False	String		14	mmddyyyyhhMMss
RETURNSTATE	False	String		2	State Returned in
SALECODE	False	String		2	Sale Code

**CARD:**

Property Name	Required	Type	Value	Max Length	Description
CARDCODE	False	String		4	VI,MS,DS – 3; AX – 4
CARDNUMBER	False	String		24	Positive integers only, no spaces
EXPDATE	False	String		4	MMYY
KSN	False	String		19	Debit Only – KSN Number from PIN pad
PINBLOCK	False	String		16	Debit Only – PinBlock from PIN pad
TRACKDATA	False*	String		200	Track Data as read from magnetic strip. May contain Track 1 and/or Track 2 data. <a href="#">See Appendix</a>

\* Required if METHOD = DB

**CHECK:**

Property Name	Required	Type	Value	Max Length	Description
ABACODE	False	String		9	Bank Routing/ABA Number
ACCOUNTNAME	False	String		50	Account Holder Name
ACCOUNTNUM	False	String		20	Account Number
ACCOUNTTYPE	False	String	CHECKING/SAVINGS	8	Type of Bank Account
BANKNAME	False	String		50	Name of Bank
CHECKNUM	False	String		20	Check Number
MICRDATA	False	String		400	MICR Data Read from Check reader
SECCODE	False	String		3	ACH Transaction Classification <a href="#">See Appendix</a>

**CUSTOMER\_BILL:**

Property Name	Required	Type	Value	Max Length	Description
ADDRESS	False	String		60	Billing Street address
CITY	False	String		40	Billing City
COMPANY	False	String		50	Billing Company
COUNTRY	False	String		60	Billing Country
EMAIL	False	String		255	Billing Email address
EMAILRECEIPT	False	String	TRUE/FALSE	5	If no email is provided value is considered FALSE
FIRSTNAME	False	String		50	Billing First Name
LASTNAME	False	String		50	Billing Last Name
PHONE	False	String		25	Billing Phone Number
STATE	False	String		40	Billing State
ZIP	False	String		20	Billing Zip Code

**CUSTOMER\_SHIP:**

Property Name	Required	Type	Value	Max Length	Description
ADDRESS	False	String		60	Shipping Street address
CITY	False	String		40	Shipping City
COMPANY	False	String		50	Shipping Company
COUNTRY	False	String		60	Shipping Country
FIRSTNAME	False	String		50	Shipping First Name
LASTNAME	False	String		50	Shipping Last Name
STATE	False	String		40	Shipping State
ZIP	False	String		20	Shipping Zip Code

**HOTEL:**

Property Name	Required	Type	Value	Max Length	Description
CHARGETYPE	False	String		1	Type of Charge
CHECKIN_DATE	False	String		10	mmddyyyy
CHECKOUT_DATE	False	String		10	mmddyyyy
EXTRA_CHARGEAMOUNT	False	Decimal		15	Additional Charge Amount
EXTRA_CHARGECODE	False	String		10	Additional Charge Code
PREFERRED_CUSTOMER	False	String	Y/N	1	
ROOMRATE	False	Decimal		15	Rate of room
SALECODE	False	String		2	Sale Code

**LEVEL2:**

Property Name	Required	Type	Value	Max Length	Description
DUTY	False	Decimal		15	Duty Amount
FREIGHT	False	Decimal		15	Freight Amount
PONUM	False	String		50	Purchase Order Number
TAX	False	Decimal		15	Tax Amount
TAXFLAG	False	Integer	0/1/2	1	0 = no tax 1 = Tax Included 2 = Tax Exempt

**LEVEL3:**

Property Name	Required	Type	Value	Max Length	Description
ALTTAX_AMT	False	Decimal		15	Alternate tax Amount
ALTTAX_AMT_INDICATOR	False	String	Y/N	1	Alternate tax indicator
COUNTRYCODETO	False	String		3	Country code shipped to
DUTYAMT	False	Decimal		15	Duty Amount
FREIGHTAMT	False	Decimal		15	Freight Amount
MERCHANT_VAT_REGISTRATION_NUM	False	String		20	Merchant VAT Registration Number
NATIONALTAX_AMT	False	Decimal		15	National Tax Amount
NATIONALTAX_AMT_INDICATOR	False	String	Y/N	1	National Tax Amount Indicator
ORDER_DISCOUNT_AMT	False	Decimal		15	Order Discount Amount
PURCHASER_ORDER_DATE	False	String		10	Purchase Order Date
PURCHASER_VAT_REGISTRATION_NUM	False	String		20	Purchaser VAT Registration Number
SUMMARY_COMMODITY_CODE	False	String		4	Summary Commodity Code
VATTAX_AMT	False	Decimal		15	VAT Tax Amount
VATTAX_RATE	False	Decimal		15	VAT Tax Rate
VAT_INVOICE_REFNUM	False	String		15	VAT invoice Reference Number
ZIPFROM	False	String		20	Zip Code Shipped From
ZIPTO	False	String		20	Zip Code Shipped To
LEVEL3PRODUCTS	False	Object Array			Array of LEVEL3PRODUCT

**LEVEL3PRODUCT:**

Property Name	Required	Type	Value	Max Length	Description
ALTTAXID	False	String		15	Alternate tax Amount
COMMODITYCODE	False	String		12	Commodity Code
DISCOUNTAMT	False	Decimal		15	Discount Amount
DISCOUNTRATE	False	Decimal		15	Discount Rate
DISCOUNT_INDICATOR	False	String	Y/N	1	Discount Indicator
GROSSNET_INDICATOR	False	String	Y/N	1	Gross Net Indicator
ITEMCODE	False	String		12	Item Code
ITEMNAME	False	String		35	Item Name
ITEMTOTALAMT	False	Decimal		15	Item Total Amount
QUANTITY	False	Decimal		15	Item Quantity
TAXAMT	False	Decimal		15	Tax Amount
TAXRATE	False	Decimal		15	Tax Rate
TAXTYPE_APPLIED	False	String		4	Tax Type that was Applied
UNIT	False	String		12	Unit of Measure
UNITPRICE	False	Decimal		15	Unit Price

**MERCHANT\_KEY – REQUIRED FOR ALL TRANSACTIONS**

Property Name	Required	Type	Value	Max Length	Description
GROUPID	False	Integer		15	SecureNet Group ID  *not applicable for all merchants should be provided by SecureNet
SECURENETID	True	Integer		7	7 Digit SecureNet ID
SECUREKEY	True	String		50	Case Sensitive SecureNet SecureKey

**MPI:**

Property Name	Required	Type	Value	Max Length	Description
AUTHINDICATOR	False	String		50	The electronic commerce indicator (ECI) value for a Visa transaction; or the universal cardholder authentication field (UCAF) indicator for MasterCard transaction.

Property Name	Required	Type	Value	Max Length	Description
AUTHVALUE	False	String		50	The cardholder authentication verification value (CAVV) for Visa transactions; or accountholder authentication value (AVV)/ universal cardholder authentication field (UCAF) for MasterCard transactions.

## PETROLEUM:

Property Name	Required	Type	Value	Max Length	Description
DRIVEJOBNUM	False	String		6	Driver Job Number
FLEET_REFERENCENUM	False	String		10	Fleet Reference Number
FULLSERV_INDICATOR	False	String		3	Full Service indicator
GALLONS	False	Integer		15	Number of Gallons
ODOMETER	False	String		6	Odometer reading
PIN	False	String		10	
TAXAMOUNT	False	Decimal		15	Tax Amount
UNITPRICE	False	Decimal		15	Price per unit
VEHICLENUM	False	String		6	Vehicle Number
PRODUCTS	False	Object Array			Product Listing

## PRODUCT:

Property Name	Required	Type	Value	Max Length	Description
AMOUNT	False	Decimal		15	Product Amount
CODE	False	String		9	Product Code
DESCRIPTION	False	String		50	Product description
NAME	False	String		30	Product Name
QUANTITY	False	String		10	Product Quantity
TAX	False	Decimal		15	Product Tax Amount
TAXABLE	False	String	1/0	1	0 = No 1 = Yes
UNIT	False	String		9	Unit of Measure
UNITPRICE	False	DECIMAL		15	Price per Unit



**SERVICE:**

Property Name	Required	Type	Value	Max Length	Description
GRATUITY	False	Decimal		15	Gratuity Amount
SERVERNUM	False	String		6	Server Number

**USERDEFINED:**

Property Name	Required	Type	Value	Max Length	Description
UD1	False	String		50	Additional Data Field
UD2	False	String		50	Additional Data Field
....	False	String		50	Additional Data Field
UD50	False	String		50	Additional Data Field

**VAULT OBJECTS****TRANSACTION\_VAULT**

Property Name	Required	Type	Value	Max Length	Description
ACCOUNT_VAULT	False	Object			Account information to add, update or delete
CUSTOMER_VAULT	False	Object			Customer information to add, update or delete
MERCHANT_KEY	True	Object			Merchant_Key is required to process any operation
OPERATIONPARAMETERS	False	<b>Object</b>			To specify if it's a Add/Update or Delete operation and to specify if Vault information to be added even if transaction declined in ProcessVaultTransaction method
TRANSACTION	False	<b>Object</b>			Transaction information for ProcessVaultTransaction method

## ACCOUNT\_VAULT

Property Name	Required	Type	Value	Max Length	Description
ACDI	False	Integer	0, 1, 2	1	Account Duplicate Indicator <a href="#">See Appendix</a>
CUSTOMERID	True	String	Any String	25	Contains the customer ID associated with the transaction or record
METHOD	True	String	CC, ECHECK, PD	7	Indicates the method of payment for the transaction being sent to the system. If left blank, defaults to CC.  PD: PIN-less Debit*
NOTES	False	String		500	Additional Notes field
PAYMENTID	True	String	Any String or AUTO	25	Contains the Payment ID associated with the customer record.  If AUTO is specified, a unique PAYMENTID is generated and returned in the response object.
PRIMARY	False	String	TRUE, FALSE	5	Specifies if the account is Primary or not
CARD	False	<b>OBJECT</b>			Card Payment Object
CHECK	False	<b>OBJECT</b>			Check Payment Object
CUSTOMER_BILL	False	<b>OBJECT</b>			Object containing Billing Customer information
USERDEFINED	False	<b>OBJECT</b>			Object Containing 50 user defined fields

## CUSTOMER\_VAULT

Property Name	Required	Type	Value	Max Length	Description
CSDI	True	Integer	0,1	1	Customer Duplicate Check Indicator  0 – If Customer ID Exist Return an Error  1 – if Customer ID exists do not add and continue with transaction
CUSTOMERID	True	String	Any String or AUTO	25	Contains the customer ID associated with the transaction or record.  If AUTO is specified a unique CUSTOMERID is generated and returned in the response object.
NOTES	False	String		200	Additional Notes field
CUSTOMER_BILL	False	<b>Object</b>			Object containing Billing Customer information
USERDEFINED	False	<b>Object</b>			Object Containing 50 user defined fields

## OPERATIONPARAMETERS

Property Name	Required	Type	Value	Max Length	Description
ACTIONCODE	False	Integer	1,2,3	1	1 = ADD 2 = Update 3 = Delete
ADD_IF_DECLINED	False	Integer	0,1	1	Add Customer/Account if transaction is declined 0 – False 1 – True

## TRANSACTIONIMAGE

Property Name	Required	Type	Value	Max Length	Description
SIGNATURE	False	Byte[]		3000	Byte array from signature reader
TRANSACTIONID	False	Integer		15	Transaction ID of referenced transaction
CHECKIMAGE	False	Object			CHECKIMAGE Object
MERCHANTKEY	False	Object			Object used to verify merchant

## CHECKIMAGE

Property Name	Required	Type	Value	Max Length	Description
BACKSIDE	False	Byte []		3000	Byte array from check image reader
FRONTSIDE	False	Byte []		3000	Byte array from check image reader

## GATEWAYRESPONSE

This section describes the response returned by the gateway when a merchant server submits a transaction for processing. The response is an object that contains properties and response values showing the status of a transaction. The merchant server retrieves this data and determines the message to display to the end-user.

## GATEWAYRESPONSE

Property Name	Type	Value	Max Length	Description
ASPREPONSE	Object			Response for ASP Transactions
TRANSACTIONRESPONSE	Object			Response for General Transactions
VAULTACCOUNTRESPONSE	Object			Response for SecureNet Vault Account Transactions
VAULTCUSTOMERRESPONSE	Object			Response for SecureNet Vault Customer Transactions

## TRANSACTIONRESPONSE

Property Name	Type	Value	Max Length	Description
ADDITIONAL_AMOUNT	Decimal		15	Future Use
ADDITIONALDATA1	String		100	Contains surcharge for debit transactions, if one has been setup on the Host. (If surcharge applies, it must be printed on the customer receipt.)
ADDITIONALDATA2	String		100	Contains Trace Number for debit transactions. Optional data returned by the Host for Fleet Cards. (If data exists, it must be printed on the customer receipt as reference data.)
ADDITIONALDATA3	String		100	Future Use
ADDITIONALDATA4	String		100	Future Use
ADDITIONALDATA5	String		100	Future Use
AUTHCODE	String		6	The six-digit alphanumeric authorization or approval code.
<b>AUTHORIZED_AMOUNT</b>	Decimal		15	Authorized Transaction Amount.
AVS_RESULT_CODE	String		2	Address Verification System Result Code <a href="#">See Appendix</a>
BANK_ACCOUNTNAME	String		50	Name of Account holder
BANK_ACCOUNTTYPE	String		8	Echoed from TRANSACTION
BATCHID	String		15	Default = 0, otherwise the Batch number of the transaction
CARD_CODE_RESPONSE_CODE	String		2	Response code from Credit Card Identification Code verification. <a href="#">See Appendix</a>
CARDHOLDER_FIRSTNAME	String		50	First Name of Card Holder(Track Data required)
CARDHOLDER_LASTNAME	String		50	Last Name of Card Holder (Track Data required)
CARDLEVEL_RESULTS	String		10	Additional information about the credit card type. <a href="#">See Appendix</a>
CARDTYPE	String		10	Card Type Code is returned. <a href="#">See Appendix</a>
CASHBACK_AMOUNT	Decimal		15	Echoed from TRANSACTION
CAVV_RESPONSE_CODE	String		2	Indicates the results of Cardholder Authentication Verification Value (CAVV) verification <a href="#">See Appendix</a>
CHECKNUM	String		20	Echoed from TRANSACTION
CODE	String		4	Echoed from TRANSACTION
CUSTOMERID	String		25	Echoed from TRANSACTION
EXPIRYDATE	String		4	Echoed from TRANSACTION

Property Name	Type	Value	Max Length	Description
GRATUITY	Decimal		15	Echoed from TRANSACTION
INDUSTRYSPECIFICDATA	String		1	Echoed from TRANSACTION
LAST4DIGITS	String		4	Last 4 digits of card or bank account number
LEVEL2_VALID	String	TRUE/ FALSE	5	Is Level 2 valid
LEVEL3_VALID	String	TRUE/ FALSE	5	Is Level 3 valid
MARKETSPECIFICDATA	String		1	Echoed from TRANSACTION
METHOD	String		7	Echoed from TRANSACTION
NETWORKCODE	String		10	Network Code for DB transactions
NETWORKID	String		10	Transaction Identifier for Auth Network
ORDERID	String		25	Echoed from TRANSACTION
PAYMENTID	String		25	Echoed from TRANSACTION
RESPONSE_CODE	String		10	Transaction Response Code <a href="#">See Appendix</a>
RESPONSE_REASON_CODE	String		10	Code for providing more details about the result of the transaction.
RESPONSE_REASON_TEXT	String		200	Brief description of the result, which corresponds with the Response Reason Code.
RESPONSE_SUBCODE	String		10	Code used by the system for internal transaction tracking.
RETEREFERENCENUM	String		50	Bank authorization reference #
SECURENETID	Integer		7	Echoed from TRANSACTION
SETTLEMENT_AMOUNT	Decimal		15	Settlement amount of Transaction
SETTLEMENTDATETIME	String		14	Settlement date of transaction(MMDDYYYYHHMM00)
SYSTEM_TRACENUM	String		50	System Trace # for Auth Network
TRACKTYPE	String	0,1,2	1	Track Type used for transaction, 0 if no track data is present
TRANSACTIONAMOUNT	Decimal		15	Echoed from TRANSACTION
TRANSACTIONDATETIME	String		14	Transaction date and time as recorded in the gateway according to merchant time zone. (MMDDYYYYHHMMSS)
TRANSACTIONID	Integer		15	System Generated Unique ID for transaction. Used to submit a modification of this transaction at a later time, such as voiding, crediting or capturing the transaction.
CUSTOMER_BILL	Object			Echoed from TRANSACTION

## VAULTACCOUNTRESPONSE

Property Name	Type	Value	Max Length	Description
ACCOUNTTYPE	String		8	Echoed from ACCOUNT_VAULT
ADDITIONALDATA1	String		100	Future Use
ADDITIONALDATA2	String		100	Future Use
CARDTYPE	String		10	Card Type Code is returned. <a href="#">See Appendix</a>
COMPANY	String		50	Echoed from ACCOUNT_VAULT
CUSTOMERID	String		25	Contains the customer ID associated with the transaction or record
FIRSTNAME	String		50	Echoed from ACCOUNT_VAULT
LAST4DIGITS	String		4	Last 4 digits of card or bank account
LASTNAME	String		50	Echoed from ACCOUNT_VAULT
METHOD	String		7	Echoed from ACCOUNT_VAULT
PAYMENTID	String		25	Contains the Payment ID associated with the customer record.
RESPONSE_CODE	String		10	Transaction Response Code <a href="#">See Appendix</a>
RESPONSE_REASON_CODE	String		10	Code for providing more details about the result of the transaction.
RESPONSE_REASON_TEXT	String		200	Brief description of the result, which corresponds with the Response Reason Code.
RESPONSE_SUBCODE	String		10	Code used by the system for internal transaction tracking.
TRANSACTIONDATETIME	String		14	Transaction date and time as recorded in the gateway according to merchant time zone. (MMDDYYYYHHMMSS)

## VAULTCUSTOMERRESPONSE

Field	Type	Value	Max Length	Description
COMPANY	String		50	Echoed from CUSTOMER_VAULT
CUSTOMERID	String		25	Contains the customer ID associated with the transaction or record
FIRSTNAME	String		50	Echoed from CUSTOMER_VAULT
LASTNAME	String		50	Echoed from CUSTOMER_VAULT
RESPONSE_CODE	String		10	Transaction Response Code <a href="#">See Appendix</a>
RESPONSE_REASON_CODE	String		10	Code for providing more details about the result of the transaction.
RESPONSE_REASON_TEXT	String		200	Brief description of the result, which corresponds with the Response Reason Code.
RESPONSE_SUBCODE	String		10	Code used by the system for internal transaction tracking.
TRANSACTIONDATETIME	String		14	Transaction date and time as recorded in the gateway according to merchant time zone. (MMDDYYYYHHMMSS)

## Method Overview

MethodName	Operation	Description
ProcessTransaction	Non Vault Transaction	Parameters: TRANSACTION Method processes non Vault Transactions with ProcessTransbyCID default = 0
ProcessAccount	AddAccount	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>1</b> for <b>add</b> operation. All account information to be added should be passed inside ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object
ProcessAccount	UpdateAccount	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>2</b> for <b>update</b> operation. PaymentId, CustomerId should be passed inside ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object
ProcessAccount	DeleteAccount	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>3</b> for <b>delete</b> operation. PaymentId, CustomerId and should be passed inside ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object
ProcessCustomer	AddCustomer	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>1</b> for <b>add</b> operation. All customer information to be added should be passed inside CUSTOMER_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object

MethodName	Operation	Description
ProcessCustomer	UpdateCustomer	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>2</b> for <b>update</b> operation. CustomerId and MERCHANT_KEY should be passed inside ACCOUNT_VAULT and TRANSACTION_VAULT object respectively.
ProcessCustomer	DeleteCustomer	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>3</b> for <b>delete</b> operation. CustomerId should be passed inside CUSTOMER_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object
ProcessCustomerAndAccount	AddCustomerAndAccount	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>1</b> for <b>add</b> operation. All customer information to be added should be passed inside CUSTOMER_VAULT object and account information inside ACCOUNT_VAULT and MERCHANT_KEY inside TRANSACTION_VAULT object
ProcessCustomerAndAccount	UpdateCustomerAndAccount	Method accepts TRANSACTION_VAULT. The ACTIONCODE inside OPERATIONPARAMETERS object should be set to <b>2</b> for <b>update</b> operation. All customer information to be added should be passed inside CUSTOMER_VAULT object and account information inside ACCOUNT_VAULT and MERCHANT_KEY inside TRANSACTION_VAULT object



MethodName	Operation	Description
ProcessTransaction	ProcessTransbyCID	<p>Method to process transaction on-demand for the stored Customer Record. TRANSACTION_SERVICE property value inside TRANSACTION object decides the type of operation to be performed.</p> <p>Regular = 0 , ProcessTransbyCID = 1, ProcessTransbySecondaryID= 2 , ProcessTransNewCIAI = 3</p> <p>MERCHANT_KEY should be supplied within TRANSACTION object</p>
ProcessTransaction	ProcessTransbySecondaryID	<p>Method to process transaction on-demand for the stored Customer Record. TRANSACTION_SERVICE property value inside TRANSACTION object decides the type of operation to be performed.</p> <p>Regular = 0 , ProcessTransbyCID = 1, ProcessTransbySecondaryID= 2 , ProcessTransNewCIAI = 3</p> <p>MERCHANT_KEY should be supplied within TRANSACTION object and Secondary merchant key should be supplied inside SECONDARY_MERCHANT_KEY property of TRANSACTION object</p>
ProcessVaultTransaction	ProcessTransNewCIAI	<p>Method to process transaction on-demand and to store a new Customer and account to Vault.</p> <p>TRANSACTION_SERVICE property value inside TRANSACTION object decides the type of operation to be performed.</p> <p>Regular = 0 , ProcessTransbyCID = 1, ProcessTransbySecondaryID= 2 ProcessTransNewCIAI = 3</p> <p>MERCHANT_KEY should be supplied within TRANSACTION_VAULT object , customer information to be added inside CUSTOMER_VAULT, account information to be added inside ACCOUNT_VAULT and transaction information except MERCHANT_KEY inside TRANSACTION</p>

## Appendix

**ACDI:** [Account Duplicate Check Indicator] Indicates whether duplicate account exists.

- 0 – Checks for Duplicate Card Number for specified Customer ID
- 1 – Checks for Duplicate Card Number and Expiration Date for specified Customer ID
- 2 – Checks for Duplicate Card Number for All Customer IDs for specified SecureNet ID
- 3 – Checks for Duplicate Card Number for All Customer IDs for specified Group ID

**Note:** ACCOUNT\_VAULT Only

**AVS\_RESULT\_CODE:** Indicates the result of Address Verification System (AVS) checks

### Address Verification System

The Address Verification System (AVS) helps merchants detect suspicious transaction activity. To use this system, the merchant must submit the customer's credit card billing address (numeric value only) 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.

AVS Code	Description
0	AVS data not provided
A	Street address matches, Zip Code does not
B	Postal code not verified due to incompatible formats
C	Street address and postal code not verified due to incompatible formats
D	Street address and postal code match
E	AVS data is invalid
G	Non-U.S. issuing bank does not support AVS
I	Address information not verified by international issuer
M	Customer Name, Billing Address and Zip match
N	Neither street address nor Zip code match
P	Street address not verified due to incompatible format
R	Retry: issuer's system unavailable or timed-out
S	U.S. issuing bank does not support AVS
T	Street address does not match, but 9-digit Zip code matches
U	Address information is unavailable
W	9-digit Zip matches, street address does not
X	Street address and 9-digit Zip match
Y	Street address and 5-digit Zip match
Z	5-digit Zip matches, street address 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.

**CARD\_CODE\_RESPONSE\_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. The gateway passes back this response code to the merchant.

Code	Description
M	Match
N	No match
P	Not processed
S	Data not present
U	Issuer unable to process request
Y	Card Code Matches (Amex Only)

**CARDLEVEL\_RESULTS:** Additional information on the type of card

- A - Visa Traditional
- B - Visa Traditional Rewards
- C - Visa Signature
- D - Visa Infinite
- G - Visa Business Card
- I - Visa Commerce
- K - Visa Corporate Card
- M - MasterCard/EuroCard and Diners
- Q - Private Label
- R - Proprietary Card
- S - Visa Purchasing Card
- U - Visa TravelMoney
- G1 - Visa Signature Business
- S1 - Visa Fleet
- AX1 - American Express
- DI1 - Discover
- H - Classic Visa Check Card or Prepaid
- N - Platinum Visa Check Card
- P - Gold Visa Check Card
- G2 - Visa Business Check Card
- J1 - General Prepaid
- J2 - Prepaid Gift Card
- J3 - Prepaid Healthcare

**CAVV\_RESPONSE\_CODE:** Indicates the results of Cardholder Authentication Verification Value (CAVV) verification

- Blank – CAVV not validated
- 0 – CAVV not validated because erroneous data was submitted
- 1 – CAVV failed validation
- 2 – CAVV passed validation
- 3 – CAVV validation could not be performed; issuer attempt incomplete
- 4 – CAVV validation could not be performed; issuer system error
- 5 – Reserved for future use
- 6 – Reserved for future use
- 7 – CAVV attempt – failed validation – issuer available (U.S.-issued card/non-U.S acquirer)
- 8 – CAVV attempt – passed validation – issuer available (U.S.-issued card/non-U.S. acquirer)
- 9 – CAVV attempt – failed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer)
- A – CAVV attempt – passed validation – issuer unavailable (U.S.-issued card/non-U.S. acquirer)
- B – CAVV passed validation, information only, no liability shift

**CARDTYPE Codes:**

The CARDTYPE Code is a system-specific code for different types of cards as identified by the gateway.

CARDTYPE Code	Response Description
ACH	e-Check
AX	American Express
DB	Debit Card
DC	Diners Club
DS	Discover
GC	Gas Card (Fuelman)
JC	JCB
MC	MasterCard
MCF	MasterCard Fleet
PD	PIN-less Debit
SV	Stored Value Card
VI	Visa
VIF	Visa Fleet
VY	Voyager
WX	Wright Express (WEX) Card
CHECK21	Check 21

**Credit Card Transactions:**

There are two steps to process credit card transactions:

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 default cut-off time for batch settlement of credit card transactions is 10:00 PM Eastern Daylight Time. All transactions prior to the cut-off time will be sent for settlement/batch closing automatically.

**CODE:** Transaction Type Code The following table describes the Codes of transactions that can be submitted to the gateway and how the gateway will process them.

CODE	Transaction Type	Description
0000	AUTH_ONLY	Used to validate a credit/debit card for the amount of goods sold. 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.
0001	PARTIAL_AUTH_ONLY	Used to validate a debit card for the amount of goods sold and may return an amount less than the requested amount if the total amount cannot be fully authorized. 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.
0100	AUTH_CAPTURE	Identical to AUTH_ONLY, however on approval the transaction will be picked up for settlement.
0101	PARTIAL_AUTH_CAPTURE	Identical to PARTIAL_AUTH_ONLY, however on approval the transaction will be picked up for settlement.
0200	PRIOR_AUTH_CAPTURE	<p>This transaction is used to request settlement for a transaction that was previously submitted as an AUTH_ONLY. The gateway will accept this transaction and initiate settlement if the following conditions are met:</p> <ul style="list-style-type: none"> <li>· The transaction is submitted with the ID of the original authorization-only transaction to be settled.</li> <li>· The transaction ID is valid and the system has a record of the original authorization-only transaction.</li> <li>· The original transaction referred to is not already settled, expired, or voided.</li> <li>· The amount being requested for settlement is less than or equal to the original authorized amount.</li> </ul> <p>If no amount is submitted in this transaction, the gateway will initiate settlement for the amount originally authorized.</p> <p><b>Note:</b> If extended line item, tax, freight, and/or duty information was submitted with the original transaction, adjusted information may be submitted in the event that the transaction amount changed. If no adjusted line item, tax, freight, and/or duty information is submitted, the information submitted with the original transaction will apply.</p>
0300	CAPTURE_ONLY	<p>This is a request to settle a transaction that was not submitted for authorization through the payment gateway. The gateway will accept this transaction if an authorization code is submitted.</p> <p>* AUTH_CODE is a required field for CAPTURE_ONLY transactions.</p>

<b>CODE</b>	<b>Transaction Type</b>	<b>Description</b>
0400	VOID	<p>This action cancels a previous transaction such that it is not sent for settlement. It can be performed on the following transaction types: CREDIT, AUTH_CAPTURE, CAPTURE_ONLY, and AUTH_ONLY. The transaction will be accepted by the gateway if the following conditions are met:</p> <ul style="list-style-type: none"> <li>· The transaction is submitted with the ID of the original transaction to be voided (REF_TRANSID).</li> <li>· The gateway has a record of the transaction referenced by the ID.</li> <li>· The transaction has not been sent for settlement.</li> </ul>
0401	PARTIAL_VOID	<p>This action will reduce the amount of a previous unsettled transaction.</p> <ul style="list-style-type: none"> <li>-The transaction is submitted with the ID of the original transaction to be voided (REF_TRANSID).</li> <li>- The gateway has a record of the transaction referenced by the ID.</li> <li>- The transaction has not been sent for settlement.</li> </ul>
0500	CREDIT	<p>This transaction is also referred to as a "Refund" and indicates to the gateway that money should flow from the merchant to the customer. The gateway will accept a credit (refund) request if the transaction submitted meets the following conditions:</p> <ul style="list-style-type: none"> <li>· The transaction is submitted with the ID of the original transaction against which the credit is being issued (REF_TRANSID).</li> <li>· The transaction ID is valid and the system has a record of the original transaction.</li> <li>· The original transaction has been settled.</li> <li>· The sum submitted in the credit transaction and all credits submitted against the original transaction is less than or equal to the original transaction amount.</li> <li>· The card number (or the last four digits thereof) submitted with the credit transaction matches the card number of the original transaction.</li> <li>· The credit transaction is submitted within 120 days of the settlement date and time of the original transaction.</li> </ul>
0501	CREDIT_AUTHONLY	Similar to CREDIT but the transaction is not eligible to be captured until it is marked by Capture using CREDIT_PRIORAUTHCAPTURE
0502	CREDIT_PRIORAUTHCAPTURE	Capture CREDIT_AUTHONLY transaction and mark for Capture and Settlement
0600	FORCE_CREDIT	Similar to CREDIT used for Unlinked credits when previous Authorized Transaction is not Available. This feature has to be enabled by Merchant Service Provider on the Gateway
0601	FORCE_CREDIT_AUTHONLY	Similar to FORCE_CREDIT but the transaction is not eligible to be captured until it is marked by Capture using FORCE_CREDIT_PRIORAUTHCAPTURE
0602	FORCE_CREDIT_PRIORAUTHCAPTURE	Capture FORCE_CREDIT_AUTHONLY transaction and mark for Capture and Settlement
0700	VERIFICATION	Run Transaction Verification on Credit Card with zero dollar amount, applicable for Visa only.

CODE	Transaction Type	Description
0800	AUTH_INCREMENT	<p>This action performs an incremental authorization on a previously authorized transaction (i.e., AUTH_ONLY). Only applicable to Visa transactions and only if the terminal has been set up as LODGING or AUTO RENTAL industry type. The transaction will be accepted by the gateway if the following conditions are met:</p> <ul style="list-style-type: none"> <li>· The transaction is submitted with the ID of the transaction.</li> <li>· The gateway has a record of the transaction referenced by the ID.</li> <li>· The transaction has not been sent for settlement.</li> </ul> <p>AUTH_INCREMENT can be submitted multiple times before submitting "PRIOR_AUTH_CAPTURE".</p>
0900	ISSUE	This action issues a new card or adds funds to an already activated stored value card; also used for refunds to a stored value card. Will be treated as credit to the card holder.
0901	ACTIVATE	This action activates a new card; only necessary if the stored value card is set to require activation.
0902	REDEEM	This action charges a card; the amount has to be within the available balance.
0903	REDEEM_PARTIAL	This action charges a card; the card will be approved only for the available balance on the card if the transaction is approved.
0904	DEACTIVATE	This action de-activates an activated card; amount should be specified as zero.
0905	REACTIVATE	This action re-activates a de-activated card.
0906	INQUIRY_BALANCE	This action checks the available balance on the stored value card; amount should be specified as zero.

**DCI** – [Duplicate Transaction Indicator] must be activated in Tools tab of Virtual Terminal: "Duplicate Transaction Lock-Out Settings". Duplicates are evaluated using AMOUNT, CARDNUMBER, and a specified time frame of only APPROVED transactions unless otherwise noted. If a duplicate transaction is found the new transaction is not reprocessed.

0 – No duplicate checking will be done

1 – if evaluation of the above conditions results in a duplicate transaction an exception code will be returned.

2 – if evaluation of the above conditions results in a duplicate transaction the original transaction response is returned with RESPONSE\_REASON\_TEXT = "Approved"

3 – Identical to 1 however ORDERID is not evaluated

**Note:** ORDERID will be evaluated and responses sent accordingly AFTER Duplicate check is evaluated and no duplicate is found

**INDUSTRYSPECIFICDATA:**

For e-Commerce transactions:

P – Physical goods

D – Digital goods

For MO/TO transactions:

1 – Single purchase transaction (AVS is required)

2 – Recurring billing transaction (do not submit AVS)

3 – Installment transaction

**MARKET\_SPECIFICDATA:**

B – Bill payment applicable for Visa only

**METHOD:** Indicates the method of payment for the transaction being sent to the system.

CC – Credit Card

ECHECK – Electronic Check

CHECK21 – Electronic Check

DB – Debit

PD – PIN-less Debit

SV – Stored Value

**ORDERID:** A client generated unique ID for each transaction submitted to the gateway. ORDERID must be unique to the SecureNet ID however the uniqueness is only evaluated for APPROVED transactions. If a transaction results in a decline that ORDERID may be used again. The ORDERID is used as a default method to avoid processing duplicate transactions. The ORDERID is not verified for transaction types such as VOID and PRIOR\_AUTH\_CAPTURE. ORDERID is limited to 25 characters; for example "CUSTOMERID MMddyyyyHHmmss".

**REF\_TRANSID:** Transaction id of previously approved transaction used in VOID and PRIOR\_AUTH\_CAPTURE

**RESPONSE\_CODE :** Indicates the result of the transaction:

1 – Approved

2 – Declined

3 – Error / Invalid Data



**SECCODE:** The Standard Entry Class (SEC) code is a three letter code that identifies the nature of the ACH entry. Here are some common SEC codes:

Code	Description
ARC	<i>Accounts Receivable Entries.</i> A check received by a merchant through mail or drop box and presented as an ACH entry.
BOC	<i>Back Office Conversion.</i> A check that are converted from paper to an electronic debit at a centralized location and presented as an ACH entry.
CCD	<i>Corporate Cash Disbursement.</i> Primarily used for business to business transactions.
POS	<i>Point-of-Sale.</i> A debit at electronic terminal initiated by use of a plastic card. An example is using your Debit card to purchase gas.
PPD	<i>Prearranged Payment and Deposits.</i> Used to credit or debit a consumer account. Popularly used for payroll direct deposits and preauthorized bill payments.
TEL	<i>Telephone Initiated-Entry.</i> Verbal authorization by telephone to issue an ACH entry such as checks by phone. (TEL code allowed for inbound telephone orders only. NACHA disallows the use of this code for outbound telephone solicitations calls. )
WEB	<i>Web Initiated-Entry.</i> Electronic authorization through the Internet to create an ACH entry.

**SOFTDESCRIPTOR:** Displays a description in addition to the merchant's DBA. This is currently restricted to credit card transactions only. Contact Support to use this field. Maximum length is 25 characters.

<sup>†</sup>The SOFTDESCRIPTOR field should contain the "Doing Business As" (DBA) name of the merchant and be the name most recognizable to the cardholder. The SOFTDESCRIPTOR field must not be used as a description field in lieu of the required, recognizable name. In addition, the SOFTDESCRIPTOR field may contain a merchandise description, order number, reference number, or other information that will further identify the transaction and assist cardholder recognition. If additional information is used, the SOFTDESCRIPTOR field must conform to one of the following formats:

	Field Position	Data
<i>Option 1:</i>	Pos. 1–3:	Merchant name or abbreviation
	Pos. 4:	Asterisk (*)
	Pos. 5–25:	Descriptive Information
<i>Option 2:</i>	Pos. 1–7:	Merchant name or abbreviation
	Pos. 8:	Asterisk (*)
	Pos. 9–25:	Descriptive Information

<i>Option 3:</i>	Pos. 1–12:	Merchant name or abbreviation
	Pos. 13:	Asterisk (*)
	Pos. 14–25:	Descriptive Information

**TRACKDATA:** This field contains the information encoded from a valid Track magnetic stripe read. It includes information such as the Primary Account Number and Expiration Date. Entire Track Data must be forwarded intact. On Debit Card transactions, this field is mandatory (with the exception of Debit Card PIN-Less transactions). Debit Card PIN-Less transactions are manually entered transactions.

**TRANSACTION\_SERVICE:**

[Default]

0 – Regular Transaction

[SecureNet Vault Transactions]

1 – Process Transaction using Customer ID

2 – Process Transaction using SECONDARY\_MERCHANT\_KEY

3 – Process Transaction Add Customer and Account

## Petroleum Industry Transaction Data – Product Codes

### Paymentech Petroleum Product Codes/Paymentech 'Generic' Codes

Ethan1 (Regular)	001	Wipers	023
Diesel	002	Battery	024
Unleaded	003	Fluids/Coolants	025
Super Unleaded	004	Tires	026
Other Fuel	005	Hoses	027
Unleaded Plus	006	Filters	028
Gas/Alcohol	007	Repairs	030
Gas/Alcohol 2	008	Oil Change	031
Diesel 2	009	Tune Up	032
Propane	010	Inspection	033
CNG	011	Car Wash	034
Unleaded 10% blend	012	Tires/Batteries/Access	035
Unleaded Plus 10% blend	013	Other	040
Super Unleaded 10%	014	Accessories	041
Unleaded 7% blend	015	Food	042
Unleaded Plus 7%	016	Service	043
Super Unleaded 7% blend	017	Parts	044
Refer Fuel	018	Labor	045
Farm Fuel	019	Reserved	046
Multi-fuel	020	Discount*	051
Motor Oil	021	Sales Tax**	056
Lamps	022		

\*Product Code 051: Discount is a negative amount

\*\*If tax is submitted as a Product Code, it must ALSO be submitted in the purchasing card data fields.

Product Amounts can be treated as negative values if they have been setup as such with the host. When submitted, the total of the positive product amounts minus the negative product amounts must equal the transaction amount.

Transaction amounts must cross foot:

- a) Sum of all Product Amounts must equal Transaction Amount.
- b) If provided: Unit Price \* Quantity = Product Amount (Quantity multiplied by Unit Price is rounded using 4/5 rounding to derive the extended amount.)

## Test Data

To perform tests in our test environment make sure to set Virtual Terminal to 'LIVE MODE', the 'TEST' parameter as 'FALSE', and send the transactions to <https://certify.securenet.com/API/Gateway.svc>

To request a test account, please contact your sales agent, or send a request to [support@securenet.com](mailto:support@securenet.com).

### Test Credit Card Numbers for Approved Transactions

American Express	370000000000002
Discover	6011000000000012
MasterCard	5424000000000015
Visa	40070000000027

### Fraud Prevention Settings

AVS Match	20008
CVV/CID Approval	568
CVV/CID Visa	999
CVV/CID MasterCard	998

### Test Credit Card Numbers for Declined Transactions

MasterCard	5105105105105100
MasterCard	5555555555554444
Visa	4111111111111111
Visa	4012888888881881
American Express	378282246310005
American Express	371449635398431

### Valid Routing Numbers

222371863  
307075259  
052000113

## Sample Code (C#)

Assuming the developer has service reference to the Gateway named “GatewayService”

```
using ProjectNamespace.GatewayService;
private void ProcessGatewayTransaction()
{
    //First define the primary objects that will be needed
    GatewayClient Client = new GatewayClient("wsBinding");
    TRANSACTION oT = new TRANSACTION();
    GATEWAYRESPONSE oGr = new GATEWAYRESPONSE();
    //All Gateway Transactions require a MERCHANT_KEY
    oT.MERCHANT_KEY = new MERCHANT_KEY();
    //Create an instance of a payment object
    oT.CARD = new CARD();
    //OR
    oT.CHECK = new CHECK();
    //Add some additional objects to the transaction
    oT.CUSTOMER_BILL = new CUSTOMER_BILL();
    oT.USERDEFINED = new USERDEFINED();
    //Assign some values
    oT.MERCHANT_KEY.SECURENETID = 1234567;
    oT.MERCHANT_KEY.SECUREKEY = "OurSecureKey";

    oT.CARD.CARDNUMBER = "4111111111111111";
    oT.CARD.CARDCODE = "123";
    oT.CARD.EXPDATE = "1212"; //MMYY
    oT.CUSTOMER_BILL.FIRSTNAME = "Customer";
    oT.CUSTOMER_BILL.LASTNAME = "Last Name";
    oT.USERDEFINED.UD1 = "Userdefined Value";
    oT.USERDEFINED.UD2 = "There are 50 of these to use";
    oT.AMOUNT = 1;
    oT.TEST = "FALSE"; // When testing, use TRUE
    oT.METHOD = "CC"; // We'll use a credit card
    oT.ORDERID = GetOrderID(); // Define a unique id for each transaction
    oT.CODE = "0000"; // An Auth only transaction
    //Process the Transaction
    oGr = Client.ProcessTransaction(oT);
    //Close the Client
    Client.Close();
    if (oGr.TRANSACTIONRESPONSE.RESPONSE_CODE == "1")
    {
        //Handle approved transaction
    }
    else if (oGr.TRANSACTIONRESPONSE.RESPONSE_CODE == "2")
    {
        //Handle declined transaction
    }
    else
    {
        //Handle transaction error
    }
}
```

## SecureNet Vault

### Account:

```
GatewayService.GatewayClient Client = new GatewayService.GatewayClient("wsBinding");
TRANSACTION_VAULT oTV = new TRANSACTION_VAULT();

oTV.ACCOUNT_VAULT = new ACCOUNT_VAULT();
oTV.MERCHANT_KEY = new MERCHANT_KEY();
oTV.CUSTOMER_BILL = new CUSTOMER_BILL();
oTV.CARD = new CARD();
oTV.OPERATIONPARAMETERS = new OPERATIONPARAMETERS();

oTV.OPERATIONPARAMETERS.ACTIONCODE = 1; //1 = ADD, 2 = UPDATE, 3 = DELETE
oTV.MERCHANT_KEY.SECUREKEY = "OurSecureKey";
oTV.MERCHANT_KEY.SECURENETID = 1234567;
oTV.ACCOUNT_VAULT.CUSTOMER_ID = "123";
oTV.ACCOUNT_VAULT.PAYMENT_ID = "1";
oTV.ACCOUNT_VAULT.PRIMARY = "TRUE";
oTV.ACCOUNT_VAULT.CUSTOMER_BILL.EMAIL = "Customer@SecureNet.com";
oTV.ACCOUNT_VAULT.METHOD = "CC";
oTV.ACCOUNT_VAULT.CARD.CARDNUMBER = "4111111111111111";
oTV.ACCOUNT_VAULT.CARD.EXPDATE = "1212";
oTV.ACCOUNT_VAULT.CARD.CARDCODE = "123";
oTV.ACCOUNT_VAULT.ACDI = 1;

GATEWAYRESPONSE oGr = Client.ProcessAccount(oV);
Client.Close();
if (oGr.VAULTACCOUNTRESPONSE.RESPONSE_CODE == "1")
{
    //Handle approved transaction
}
else
{
    //Handle transaction error
}
```

## Customer:

```
GatewayService.GatewayClient Client = new GatewayService.GatewayClient("wsBinding");
TRANSACTION_VAULT oTV = new TRANSACTION_VAULT();

oTV.CUSTOMER_VAULT = new CUSTOMER_VAULT();
oTV.MERCHANT_KEY = new MERCHANT_KEY();
oTV.CUSTOMER_BILL = new CUSTOMER_BILL();
oTV.CARD = new CARD();
oTV.OPERATIONPARAMETERS = new OPERATIONPARAMETERS();

oTV.OPERATIONPARAMETERS.ACTIONCODE = 1; //1 = ADD, 2 = UPDATE, 3 = DELETE
oTV.MERCHANT_KEY.SECUREKEY = "OurSecureKey";
oTV.MERCHANT_KEY.SECURENETID = 1234567;
oTV.CUSTOMER_VAULT.CUSTOMER_ID = "123";
oTV.CUSTOMER_VAULT.CUSTOMER_BILL.EMAIL = "Customer@SecureNet.com";
oTV.CUSTOMER_VAULT.CUSTOMER_BILL.FIRSTNAME = "Customer";
oTV.CUSTOMER_VAULT.CUSTOMER_BILL.LASTNAME = "Last Name";

oTV.CUSTOMER_VAULT.CSDI = 1;

GATEWAYRESPONSE oGr = Client.ProcessCustomer(oV);
Client.Close();
if (oGr.VAULTCUSTOMERRESPONSE.RESPONSE_CODE == "1")
{
    //Handle approved transaction
}
else
{
    //Handle transaction error
}
```