SecureNet Gateway API Implementation Guide

Version 4.1.4—09.14.2012

SecureNet
Payment Systems

Table of Contents

APPENDIX

1 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 SecureNet Gateway (Gateway) acts as a bridge between the merchants 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.

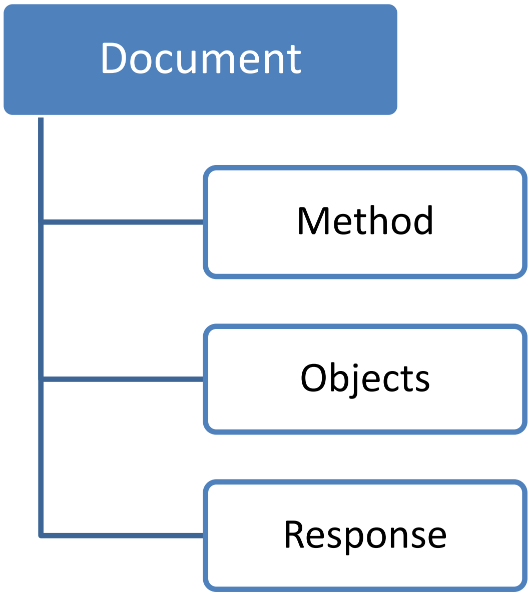
Authorization is the process of checking the validity and available balance of a customer's credit card before the transaction can be accepted. To authorize a given credit card transaction, the Gateway transmits the transaction information to the appropriate financial institutions for validation, then 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.

1.1 How to Use This Document

This document contains hyperlinks which refer the user to the corresponding sections of the document that apply. Please note that some object names may contain symbols such as underscores.



1.2 What is the Gateway?

The Gateway is the recommended method of submitting transactions. This method allows the merchants 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.

1.3 Service Based Authorization Process

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

1.4 How Does the Gateway Work?

When using the Gateway, a typical transaction is performed in the following way:

1. The customers program has a reference to the Gateway Service and connects securely to the merchants server to transmit payment information.
2. The merchants 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 merchants server.
5. The merchants server receives and processes the GATEWAYRESPONSE object.
6. Finally, the merchants server communicates the success (or failure) of the authorization to the customer’s web browser or program.

1.5 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. Complete the Gateway Certification Request Form.
2. Create an account in our test environment.
3. Establish a secure socket connection.
4. Provide both server and client-side encryption.
5. 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).
6. Securely store the following information in a way that it is configurable: Service URL, SecureNet ID, its corresponding SecureKey, and, optionally, transaction mode (TEST or LIVE). This information will be accessed by the script that submits transactions to the Gateway.
7. Complete the certification process by running test scripts for the transaction scenarios.

1.6 Certification Process

Once the merchant/developer/VAR and SecureNets representative have determined that using the API is the right way to integrate to the Gateway the API Team will initiate the certification process by following the next steps:

1. The merchant will receive a “Gateway Certification—Test Account Request Form” that should be completed and sent back to SecureNets representative.
2. The SecureNet representative will review the completed form and forward to the API Team (APITeam@SecureNet.com).
3. The API Team will assign a Developer ID that the merchant will need to include in the software/application being certified.

4. The API Team will provide the merchant the following:
    a. Latest API Documentation
    b. If available, Basic Code samples
    c. Developer ID
    d. Test Account information
5. Once the merchants development teams completes the coding, and are ready for certification, should send an email to the API Team (APITeam@SecureNet.com) requesting certification scripts.
6. The API Team will forward that request to the Certification Team which will at the same time, send the certification scripts to the merchants development team.
7. Once the merchants development team has ran all the scripts successfully should send an email to the API Team and the Certification Team requesting review of the scripts.
8. The Certification Team will review the scripts and provide feedback to the merchants development team.
9. Once all the scripts pass the certification process the Certification Team will issue a Certification Letter and send it to the merchant.
10. The Certification Letter will be issued for the specific software/application that was certified and will enable the merchant to move the code to a production environment using live SecureNet account(s) for payment processing.

1.7 Post Certification Process

Once the software/application has been certified the following steps and considerations should be done and/or taken into account:

1. The merchants development team should change all the variables and configuration files to point to the Gateway live environment (https://gateway.securenet.com).
2. It is recommended that, before the official release, the merchants development team should complete a few “production tests” using live data. The type and number of test should be at the discretion of the merchant and its development team.
3. Due to the nature of the payment industry SecureNet might make changes on the API and/or the services provided. SecureNet will inform, via official notification, any changes that will be implemented in future releases, if any of these changes impact the certified software/application the merchant should make the necessary code updates and re-certify the software/application following the Certification Process. The re-certification must be completed before SecureNets official release date and go live the same day of the official release date.
4. If the certified software/application should change in any way SecureNet recommends to re-certify in order to make sure all the new and existing functionality works as intended, and, the right version is listed at www.SecureNet.com if the option was selected in the certification form.

2 GATEWAY IMPLEMENTATION

This guide explains how to integrate SecureNet into the merchants website or application to process transactions over SecureNets payment Gateway. Transactions could be submitted to the Gateway for processing through:

- Structured XML HTTPS Post
- XML Web Service
- WCF Service

2.1 Service and SOAP Overview

The Gateway has implemented Windows Communication Foundation (WCF) as an underlying technology to support 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 integrates the application to the API that performs the following:

1. Securely obtain all the information needed to process a transaction.
2. Create a program to consume the Gateway service.

   NOTE: *The SecureNet Gateway will only accept transactions only on port 443 via https.*
   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.

2.2 End Points

2.2.1 Production Endpoints

Transaction API (Gateway Services)

- https://gateway.securenet.com/API/Gateway.svc/webHttp (XML post)
- https://gateway.securenet.com/API/Gateway.svc/soap (XML Web service)

Extended Transaction Functionality API (Extended Services)

- https://gateway.securenet.com/API/data/service.svc/webHttp (XML POST)
- https://gateway.securenet.com/API/data/service.svc/soap (XML Web service)

Extended Transaction Process API (Transaction Services)

- https://gateway.securenet.com/API/data/transaction.svc/webhttp (XML Post)
- https://gateway.securenet.com/API/data/transaction.svc/soap (XML Web service)

Global XSD

- https://gateway.securenet.com/api/validationobject.xsd

WSDL

- https://gateway.securenet.com/API/Gateway.svc?wsdl (Gateway Services)
- https://gateway.securenet.com/API/Data/SERVICE.svc?wsdl (Extended Services)
- https://gateway.securenet.com/API/Data/TRANSACTION.svc?wsdl (Transaction Services)

2.2.2 Certification Endpoints

Transaction API (Gateway Services)

- https://certify.securenet.com/API/Gateway.svc/webHttp (XML post)
- https://certify.securenet.com/API/Gateway.svc/soap (XML Web service)

Extended Transaction Functionality API (Extended Services)

- https://certify.securenet.com/API/data/service.svc/webHttp (XML POST)
- https://certify.securenet.com/API/data/service.svc/soap (XML Web service)

Extended Transaction Process API (Transaction Services)

- https://certify.securenet.com/API/data/transaction.svc/webhttp (XML Post)
- https://certify.securenet.com/API/data/transaction.svc/soap (XML Web service)

WSDL

- https://certify.securenet.com/API/Gateway.svc?wsdl (Gateway Services)
- https://certify.securenet.com/API/Data/SERVICE.svc?wsdl (Extended Services)
- https://certify.securenet.com/API/Data/TRANSACTION.svc?wsdl (Transaction Services)

Global XSD

- https://certify.securenet.com/api/validationobject.xsd

3 INTEGRATED PLATFORMS AND SERVICES

3.1 Platforms

Certified Platform	Services Available	Supported Environment
Ceridian	Stored Value	Retail Restaurant Moto E-Commerce Petroleum Lodging/Hotel Car Rental
Certegy	ACH verification	PPD, CCD, TEL, & WEB only
First Data Nashville	Credit Card, Pin Debit	Retail Restaurant Moto Ecommerce
PaymentTech Netconnect (Tampa)	Credit Card, Stored Value, Pin Debit	Retail Restaurant Moto Ecommerce Petroleum Lodging/Hotel Car Rental
SecureNet ACH Services	Pre-Note, ACH processing	PPD, CCD, WEB, TEL, ARC, BOC, POP, & POS
SecureNet FrontEnd	Credit Card, Pin Debit, Pin-less Debit, EBT	Retail Restaurant Moto Ecommerce
TSYS (VITAL)	Credit Card, Pin Debit	Retail Restaurant Moto Ecommerce

3.2 Feature Restrictions

Feature	Platform Availability	Other Restrictions
SecureNet Vault	All	MOTO merchants only
AutoBill	All	SecureNet Vault Service is required MOTO merchants only
ACH	SecureNet	RETAIL and MOTO merchants only
P2P	All	
E2E	All	Available for IDTECH certified and fulfilled by SecureNet hardware only
Soft Descriptor	SecureNet FrontEnd TSYS	Acquirer’s discretion

4 METHODS

4.1 Methods for Standard Payment Transactions [Gateway Services]

Method Name	Operation	Description	Objects	Reference
ProcessTransaction	This method can be used to perform the following operations: ▪ Process a regular Transaction ▪ Process Transaction by Customer ID ▪ Process Transaction by Secondary ID (SECONDARY_MERCHANT_KEY ▪ Process Transaction by Transaction ID ▪ Other related transaction calls Please refer to the TRANSACTION_SERVICE values for more information	To perform any of these operations using the Process Transaction method, the input Object(s) should be TRANSACTION, the TRANSACTION_SERVICE in the TRANSACTION object must be set and the MERCHANT_KEY should be supplied within TRANSACTION object. For the Process Transaction by Secondary ID operation, the SECONDARY_MERCHANT_KEY should be supplied inside the SECONDARY_MERCHANT_KEY property of the TRANSACTION object. No card/check information should be provided in the CARD/CHECK object. The Process Transaction by Customer ID operation will process transaction(s) on-demand for the stored Customer Record. **For Card Transactions:** The CARDCODE and ADDITIONALINFO objects, which are part of the CARD object, assigned to the TRANSACTION object, are used to process card transactions. **For ACH Transactions:** The SECCODE, CHECKNUM, and ADDITIONALINFO objects, which are part of the CHECK object, assigned to the TRANSACTION object, are used to process ACH transactions.	**Input Object(s):** TRANSACTION Output Object(s): GATEWAY RESPONSE	Available through: Gateway Services
UpdateTransaction	Used to save electronic Signature.	Store the electronic signature	**Input Object(s):** TRANSACTIONIMAGE **Output Object(s):** IMAGERESPONSE	Available through: Gateway Services

4.2 Methods for Vault Transactions [Gateway Services]

Method Name	Operation	Description	Required Objects	Reference
ProcessAccount	This method can be used to perform the following operations: ▪ Add Vault Account ▪ Update Vault Account ▪ Delete Vault Account	To perform any of the operations using the ProcessAccount method, the TRANSACTION_VAULT should be used as the input object. All account information to be added should be passed inside ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object. The vault inside the OPERATIONPARAMETERS object should be set for Add/Update/Delete.	**Input Object(s):** TRANSACTION_VAULT **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services

Method Name	Operation	Description	Required Objects	Reference
ProcessCustomer	This method can be used to perform the following operations: ▪ Add Vault Customer ▪ Update Vault Customer ▪ Delete Vault Customer	To perform any of the operations using this ProcessCustomer method, the TRANSACTION_VAULT should be used as the input object. The ACTIONCODE inside OPERATIONPARAMETERS object should be set for Add/Update/ Delete. All customer information to be added should be passed inside CUSTOMER_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object for the add operation. CUSTOMERID should be passed inside the ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object for the Update Operation.	**Required Object:** TRANSACTION_VAULT **Input Object(s):** <TRANSACTION_VAULT > CUSTOMER_VAULT ACCOUNT_VAULT MERCHANT_KEY **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services
ProcessCustomerAnd Account	Add Vault Customer And Account Update Vault Customer And Account	To perform any of these operations using the ProcessCustomerAndAccount method, the TRANSACTION_VAULT should be used as the input object. The ACTIONCODE inside OPERATIONPARAMETERS object should be set for Add/Update operation. All customer information to be added should be passed inside CUSTOMER_VAULT object, account information inside ACCOUNT_VAULT object and MERCHANT_KEY inside TRANSACTION_VAULT object.	**Input Object(s):** TRANSACTION_VAULT CUSTOMER_VAULT ACCOUNT_VAULT OPERATIONPARAMETERS MERCHANT_KEY **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services
ProcessVaultTransaction	Process Transaction Add Customer and Account	This is a method to process transaction on-demand and to store a new Customer and account in Vault. ReportingAPImethods property value inside TRANSACTION object should be set for Add MERCHANT_KEY should be supplied within TRANSACTION_VAULT object, customer information to be added should be supplied inside CUSTOMER_VAULT, account information to be added should be supplied inside ACCOUNT_VAULT and transaction information except MERCHANT_KEY should be supplied inside TRANSACTION object.A	**Input Object(s):** TRANSACTION_VAULT CUSTOMER_VAULT ACCOUNT_VAULT TRANSACTION MERCHANT_KEY **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services

Method Name	Operation	Description	Required Objects	Reference
CopyVaultAccount	CopyVaultAccount	This method allows an existing Vault account to copy from one customer to another and from one SecureNet ID to another	**Input parameters:** SOURCE_MERCHANT_KEY (MERCHANT_KEY), SOURCE_ACCOUNT_VAULT (ACCOUNT_VAULT), TARGET_MERCHANT_KEY (MERCHANT_KEY), TARGET_ACCOUNT_VAULT (ACCOUNT_VAULT), CREATE_CUSTOMER (integer, 0—do not create, 1—create), CHANGE_ACCOUNT_INFO_TO_TARGET (integer, 0—do not change target account, 1—change target account). **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services

4.3 Methods for AutoBill Transactions [Gateway Services]

Method Name	Operation	Description	Objects	Reference
AddABAccount	Add AutoBill Account	Use this method to add an AutoBill Account	**Input Object(s):** PLAN_AB **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services
UpdateABAccount	Update AutoBill Account	Use this method to update an AutoBill Account	**Input Object(s):** PLAN_AB **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services
UpdateABSchedule	Update AutoBill Schedule	Use this method to update an AutoBill Schedule	**Input Object(s):** PLAN_AB **Output Object(s):** GATEWAYRESPONSE	Available through: Gateway Services

4.4 Reporting Methods for Batch and Transaction Services [Transaction Services]

Method Name	Description	Objects	Reference
GetBatchTransactions	Method to retrieve the TRANSACTIONRESPONSE data for a particular Batch If BatchId = 0 approved transactions from the unclosed batch will be returned If BatchId = 1 transactions for the last closed batch along with original batch id will be returned For all other values in BatchId, transactions specific to that particular batch will be returned	**Input parameters:** MERCHANT_KEY and BatchId **Output parameter:** Array of TRANSACTIONRESPONSE objects	Available through: Transaction Services
GetTransactionByOrderID	Method to retrieve the TRANSACTIONRESPONSE data by Order ID when Transaction ID is not available Transactions with basic validation errors are not stored and returned in this method	**Input Object(s):** MERCHANT_KEY and OrderID **Output Object(s):** TRANSACTIONRESPONSE	Available through: Transaction Services

Method Name	Description	Objects	Reference
GetTransactionByTransactionID	Method to retrieve the TRANSACTIONRESPONSE data by Transaction ID	**Input Object(s):** MERCHANT_KEY and TransactionID **Output Object(s):** TRANSACTIONRESPONSE	Available through: Transaction Services
GetCurrentBatchID	This method will return the last closed Batch ID	**Input Object(s):** MERCHANT_KEY **Output Object(s):** BatchId	Available through: Transaction Services
GetTransactions	Method to retrieve transaction data for transactions processed during a specific period of time	**Input parameters:** MERCHANT_KEY, StartDate, EndDate, ExcludeDate, CustomerID, Amount, AmountSpecified, RBOnly **Output Object(s):** Array of TRANSACTIONRESPONSE objects	Available through: Transaction Services

4.5 Reporting Methods for SecureNet Vault and AutoBill Services [Extended Services]

Method Name	Description	Required Objects	Reference
GetABAccount	Method to retrieve an AutoBill account	**Input Object(s):** PLAN_AB **Output Object(s):** PLAN_AB	Available Through: Extended Services
GetABAccounts	Method to retrieve all AutoBill accounts of CustomerID	**Input Object(s):** MERCHANT_KEY and CustomerID **Output Object(s):** PLAN_AB	Available Through: Extended Services
GetVaultCustomers	Method to retrieve a Vault Customer	**Input Object(s):** MERCHANT_KEY and CustomerID **Output Object(s):** CUSTOMER_VAULT	Available Through: Extended Services
GetVaultAccountByCustomer	Method to retrieve Vault Accounts by Merchant Key and Customer ID	**Input Object(s):** MERCHANT_KEY and CustomerID **Output Object(s):** Array of ACCOUNT_VAULT objects	Available Through: Extended Services
GetVaultAccount	Method to retrieve a Vault Account	**Input Object(s):** MERCHANT_KEY, CustomerID, and PaymentID **Output Object(s):** ACCOUNT_VAULT	Available Through: Extended Services
GetVaultRecordByCardNo	Method to retrieve Vault Accounts by Merchant Key and Card Number	**Input Object(s):** MERCHANT_KEY and CardNumber **Output Object(s):** Array of ACCOUNT_VAULT objects	Available Through: Extended Services
GetP2PKey	Method to request the current active primary or active secondary P2P Encryption Key	**Input Object(s):** P2PREQUEST **Output Object(s):** P2PResponse	Available Through: Extended Services
ConfirmP2PKey	Method to request confirmation and activation of a new P2P Encryption key.	**Input Object(s):** P2PConfirmationRequest **Output Object(s):** P2PResponse	Available Through: Extended Services
CreateP2PKey	Method to Request a new P2P Encryption Key	**Input Object(s):** P2PRequest **Output Object(s):** P2PResponse	Available Through: Extended Services

4.6 Reporting Methods for P2P Encryption Services [Extended Services]

Method Name	Description	Required Object(s)	Reference
CreateP2PKey	Method to Request a new P2P Encryption Key	**Input Object(s):** P2PRequest **Output Object(s):** P2PResponse	Available Through: Extended Services
ConfirmP2PKey	Method to Request confirmation and activation of a new P2P	**Input Object(s):** P2PConfirmationRequest **Output Object(s):** P2PResponse	Available Through: Extended Services
GetP2PKey	Method to request the current, primary P2P Encryption Key.	**Input Object(s):** P2PRequest **Output Object(s):** P2PResponse	Available Through: Extended Services

4.7 Method for Batch Close [Gateway Services]

Method Name	Description	Required Object(s)	Reference
CloseBatch	Close out the batch for settlement based on SecureNet ID	Input Object(s): MERCHANT_KEY Output Object(s): BATCHDATA	Available through: Gateway Services

5 GATEWAY OBJECTS

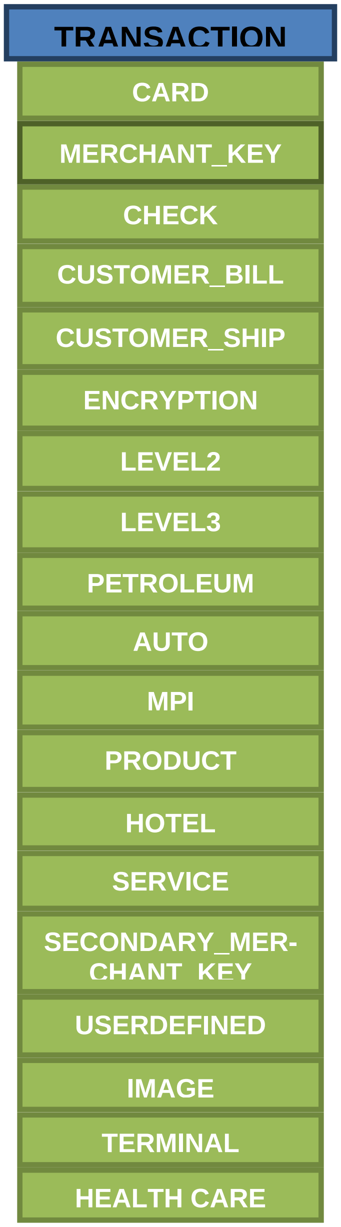
The objects submitted to the Gateway are shown below. Parameters for each object are described in this document.

5.1 Objects for Standard Transactions

This section outlines the various high level sections within the Transaction Request Object for Standard Transactions.

5.1.1 *Transaction Object Structure*

This object defines information that can be submitted to the Gateway for real-time transaction processing.



5.2 Transaction Object Specification

The parameters defined below are found in the TRANSACTION object and are assigned to the instance of this serialized object and passed to the web method "ProcessTransaction" to complete an authorization. The TRANSACTION Object can be used through Gateway Services and Transaction Services.

Object Name	Description	Constraints	Methods Used	Reference
TRANSACTION	The parameters defined below are assigned to the instance of this serialized object and passed to the web method "ProcessTransaction to complete an authorization. This object is also passed inside the TRANSACTION_VAULT object for ProcessVaultTransaction method while processing a transaction and adding vault account/customer.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
MERCHANT_KEY	This object is used to verify merchant before processing any transaction and is required for any operation.	All restrictions as defined in the Request Object Specs apply	▪ All methods	All services
CARD	This CARD payment object contains Card information like Card number, expiry date.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction ▪ ProcessAccount ▪ ProcessCustomerAndAccount ▪ CopyVaultAccount	Available Through: Gateway Services Transaction Services
CHECK	Check payment object is required for electronic check transactions and contains information corresponding checking or saving account.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction ▪ ProcessAccount ▪ ProcessCustomerAndAccount ▪ CopyVaultAccount	Available Through: Gateway Services Transaction Services
CUSTOMER_BILL	This object contains Customer Billing address and phone number and is required for Ecommerce.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction ▪ ProcessAccount ▪ ProcessCustomerAndAccount ▪ CopyVaultAccount	Available Through: Gateway Services Transaction Services
CUSTOMER_SHIP	This object contains the customers shipping information, like name and address.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction ▪ ProcessAccount ▪ ProcessCustomerAndAccount ▪ CopyVaultAccount	Available Through: Gateway Services Transaction Services
ENCRYPTION	This object is used for P2P Encryption.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services

Object Name	Description	Constraints	Methods Used	Reference
LEVEL 2	This Object is used for Level 2 data. Level2 Transactions normally involve corporate cards issued from a U.S Bank.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
LEVEL 3	This Object is used for Level 3 data. Purchases that qualify as level 3 transactions are generally made with government credit cards or corporate cards. **NOTE:** Level 3 is only available to merchants on First Data Nashville platform and only for E-COMMERCE, RETAIL and MOTO transactions.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
MPI	The Object describes Merchant Plug-in Indicator for 3D Secure Transactions.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
AUTO	This object is used for Rental Industry and includes rental-related information, like city rented, returned in and return times, etc.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
PETROLEUM	This object is used for Petroleum industry only and includes information like number of gallons, unit price, etc.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
HOTEL	The object is used for Hotel industry only and includes information, like check-in, check-out times, room rate, etc.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
PRODUCT	This object contains individual product information, like product description, amount, quantity, etc.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
SERVICE	This object is used for the Restaurant industry and contains gratuity amount and server number.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
USERDEFINED	This object contains 50 user defined fields and allows merchants to store additional data along with transaction information. **NOTE:** User defined fields are not intended to and must not be used to capture personally identifying information, like name, address, credit card number, social security number, driver's license number, state-issued identification number, passport number, and card verification numbers.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services

Object Name	Description	Constraints	Methods Used	Reference
SECONDARY_MERCHANT_KEY	This object is similar to the MERCHANT_KEY Object and is used for processing transactions across various SecureNet IDs using information from the Vault linked to the Primary SecureNet ID.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
TERMINAL	This is Object is used to provide terminal information for each transaction. It is applicable for ACH (POS SEC Code) and Ceridian Transactions.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Available Through: Gateway Services Transaction Services
HEALTHCARE	This is Object is used to provide healthcare information for each transaction.	All restrictions as defined in the Request Object Specs apply	▪ ProcessTransaction ▪ ProcessVaultTransaction	Transaction Services
IMAGE	This is Object is used to provide CHECK21 image information and Credit Card signature information.	Byte array, should be deserializable to image	▪ ProcessTransaction ▪ ProcessVaultTransaction	Transaction Services

5.3 Transaction Request Message Format

All Gateway transactions include a common set of required parameters (listed first). Additional parameters are required depending on the transaction type. Many optional parameters can also be provided depending on the results that are to be returned.

5.3.1 Transaction Request Message Format for Credit Card, Debit Card, EBT, and Stored Value Card Transactions

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<TRANSACTION>	Object / 1	True		Available Through: Gateway Services
<AMOUNT>	Decimal / 15	True	Total amount to be charged or credited. Include decimal point followed by decimal amount	
<MERCHANT_KEY>	Object / 1	True	SECURENET ID MERCHANT_KEY	Refer to MERCHANT_KEY object for additional values
<CODE>	String / 4	True	0000-AUTH_ONLY 0100-AUTH_CAPTURE 0200-PRIOR_AUTH_CAPTURE 0201-UPDATE_TRANS_INFO 0300-CAPTURE_ONLY 0400-VOID 0500- CREDIT 0600- FORCE_CREDIT	Refer to the Transaction Type Table for additional values
<METHOD>	String / 7	True	CC, DB, SV, EBT	Refer to METHOD for additional values
<CARD>	Object / 1	True		Refer to the Card Object
<OrderID>	String / 25	True	Unique Transaction Order ID. String A-Z, 0-9 only	Refer to OrderID in Appendix A

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<DCI>	Integer / 1	True	0, 1, 2, 3	Refer to DCI in Appendix A

<TEST>	String / 5	True	TRUE = Test Transaction
			FALSE = Live Transaction
			Default is FALSE
				This is case sensitive values

<TRANSACTION_SERVICE>	Integer / 1	True	Indicates Transaction Handling when Vault is enabled.	Refer to TRANSACTION_SERVICE in Appendix A
<REF_TRANSID>	String / 15	Required for Transaction Code	Transaction ID of a previously approved transaction, used in Transaction Code 0400-VOID 0401-PARTIAL_VOID 0500-CREDIT 0200-PRIOR_AUTH_CAPTURE	Refer to REF_TRANSID in Appendix A for more details Transaction id value should be number
<INDUSTRYSPECIFICDATA>	String / 1	Required for E-commerce and MOTO transactions Leave blank for retail transactions	Industry Specific Data—Required to send the right indicator for transaction. Accepted values: P, D, 1, 2, 3	Refer to INDUSTRYSPECIFICDATA in Appendix A for more information.
<INSTALLMENT_SEQUENCENUM>	Integer / 2	Required if INDUSTRYSPECIFICDATA=3	Installment number for installment payment plan 1-99	
<TOTAL_INSTALLMENTCOUNT>	Integer / 2	Required if INDUSTRYSPECIFICDATA = 3	Number of total installments for installment plans 1—99	
<AUTHCODE>	String / 6	Required for 300 transactions	Valid authorization code for Offline Transaction 0300-CAPTURE_ONLY Voice Authorization Code Valid authorization code for EBT Voucher transaction 0300-CAPTURE_ONLY	
<CUSTOMERID>	String / 25	Required for Vault transactions	Contains the customer ID associated with the Authorization Code for EBT Voucher transaction Must be enabled on the account	No space and special character
<PAYMENTID>	String / 25	Required for Vault transactions	SecureNet Vault Payment ID Must be enabled on the account	No space and special character
<AUTO>	Object / 1	Specific to Industry		Refer to the AUTO Object.
<PETROLEUM>	Object / 1	Specific to Industry		Refer to the PETROLEUM Object.
<HOTEL>	Object / 1	Specific to Industry		Refer to HOTEL Object.
<SERVICE>	Object / 1	Specific to Industry		Refer to SERVICE Object.
<CUSTOMERIP>	String / 15	False	The Customer IP address includes the Customers IP that needs to be associated with the transaction Recommended for ecommerce merchants	Valid IP of customer
<DEVICECODE>	String / 2	False	Code for payment device Blank, unless specified/ provided by SecureNet	

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<ENTRYSOURCE>	String / 10	False	Transaction Source Blank, unless specified/provided by SecureNet	
<INVOICEDESC>	String / 50	False	Description of the transaction for which invoice is being issued	
<INVOICENUM>	String / 50	Required for Ceridian	Invoice number associated with the transaction	
<MARKETSPECIFICDATA>	String / 2	False	For bill payment transactions, applicable for Visa only B—Bill payment D—Debt indicator (contact SecureNet for more info) Don't include if a value isn't sent For Nashville processor all VISA Recurring and VISA Installment transactions are considered as Bill Payment transactions by default VISA Single Bill payment MOTO transactions are also considered as Bill Payment	
<NOTE>	String / 500	False	Additional Transaction Notes	
<OVERRIDE_FROM>	Integer / 1	False	Transaction information - Billing information and UDF data - will come from the below sources based on the value of this flag: 0 - Transaction Request 1 - Customer Vault Object 2 - Account Vault Object 3 - 0200-PRIOR_AUTH_CAPTURE transaction request, Level 2 information will be overwritten with the new Level 2 values Use for 0200-PRIOR_AUTH_CAPTURE or SecureNet Vault transactions. 4 - CustomerVault Billing Information and Transaction UDF 5 - AccountVault Billing Information and Transaction UDF 6 - Transaction Billing Information CustomerVault UDF 7 - Transaction Billing Information and AccountVault UDF 8 - AccountVault BillingInformation and CustomerVault UDF 9 - CustomerVault Billing Information and AccountVault UDF	
<RETAIL_LANENUM>	String / 6	False	Lane Number for Retail stores	
<SOFTDESCRIPTOR>	String / 25	False	Additional DBA descriptor	Refer to SOFTDESCRIPTOR in the Appendix.
<CUSTOMER_BILL>	Object / 1	False		Refer to CUSTOMER_BILL Object.

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<CUSTOMER_SHIP>	Object / 1	False		Refer to CUSTOMER_SHIP Object
<ENCRYPTION>	Object / 1	False		Refer to ENCRYPTION Object
<LEVEL2>	Object / 1	False		Refer to LEVEL2 Object
<LEVEL3>	Object / 1	False		Refer to LEVEL3 Object
<PRODUCT>	Object Array	False		Refer to PRODUCT Object
<USERDEFINED>	Object / 1	False		Refer to USERDEFINED Object
<MPI>	Object / 1	False		Refer to MPI Object
<SECONDARY_MERCHANT_KEY>	Object / 1	False		Refer to SECONDARY_MERCHANT_KEY Object
<TERMINAL>	Object/1	False		Refer to Terminal Object
<DEVELOPERID>	String/8	True		Assigned by SecureNet
<VERSION>	String /10	True		Application version of the client performing the transaction. Assigned by SecureNet
<HEALTHCARE>	Object /1	False		Refer to the HEALTHCARE Object
<IMAGE>	Object /1	False		Refer to the IMAGE Object
</TRANSACTION>				

5.3.2 *Transaction Request Message Format for Electronic Check Transactions*

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<TRANSACTION>	Object / 1	True		Refer to the transaction object
<AMOUNT>	Decimal / 15	True	Total amount to be charged or credited. Include decimal point followed by decimal amount	
<MERCHANT_KEY>	Object / 1	True		Refer to the MERCHANT_KEY object for additional values
<CODE>	String / 4	True	Indicates the type of processing required for Transaction Request	Refer to the Transaction Type Table for additional values
<METHOD>	String / 7	True	Set Transaction Method to ECHECK	Refer to the METHOD for additional values in the Appendix.
<CHECK>	Object / 1	True		Refer to the Check Object
<OrderID>	String / 25	True	Unique Transaction Order ID. String A-Z, 0-9 only	Refer to the OrderID in Appendix A for additional information.
<DCI>	Integer / 1	True	Duplicate Check Indicator Sample values: 0,1,2,3	Refer to the DCI in Appendix A for additional information

XML Tag	Data Type / Max Length	Required	Sample Values	Reference

<TEST>	String / 5	True	TRUE = Test Transaction
			FALSE = Live Transaction
			Default is FALSE.
				Case sensitive value

<TRANSACTION_SERVICE>	Integer / 1	True	Indicates Transaction Handling when Vault is enabled Sample values: 0,1,2,3	Refer to the TRANSACTION_SERVICE in Appendix A for additional information
<REF_TRANSID>	String / 15	Required for 0200,0201	Transaction ID of a previously approved transaction, used in transaction codes	Refer to Appendix A for more uses of REF_TRANSID It should have number value
<PAYMENTID>	String / 25	Required for Vault transactions	SecureNet Vault Payment ID Must be enabled on the account	Alpha number value, no space and no special characters
<CUSTOMERID>	String / 25	Required for Vault transactions	Contains the customer ID associated with the transaction Must be enabled on the account	Alpha number value, no space and no special characters
<CUSTOMERIP>	String / 15	False	Customer IP address Include the Customers IP that needs to be associated with the transaction Recommended for e-Commerce merchants	
<ENTRYSOURCE>	String / 2	False	Transaction Source Blank, unless specified/provided by SecureNet	
<INVOICEDESC>	String / 50	False	Description of the transaction for which invoice is being issued	
<INVOICENUM>	String / 50	False	Invoice number associated with the transaction	
<NOTE>	String / 500	False	Additional Transaction Notes	
<OVERRIDE_FROM>	Integer / 1	False	Transaction information - Billing information and UDF data - will come from the below sources based on the value of this flag: 0 - Transaction Request 1 - Customer Vault Object 2 - Account Vault Object 3 - 0200-PRIOR_AUTH_CAPTURE transaction request, Level 2 information will be overwritten with the new Level 2 values. Use for 0200-PRIOR_AUTH_CAPTURE or SecureNet Vault transactions 4 - CustomerVault Billing Information and Transaction UDF 5 - AccountVault Billing Information and Transaction UDF 6 - Transaction Billing Information CustomerVault UDF 7 - Transaction Billing Information and AccountVault UDF 8 - AccountVault BillingInformation and CustomerVault UDF 9 - CustomerVault Billing Information and AccountVault UDF	

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<RETAIL_LANENUM>	String / 6	False	Lane Number for Retail stores	
<SOFTDESCRIPTOR>	String / 25	False	Additional DBA descriptor	Refer to the SOFTDESCRIPTOR in the Appendix
<CUSTOMER_BILL>	Object / 1	False		Refer to the CUSTOMER_BILL Object
<CUSTOMER_SHIP>	Object / 1	False		Refer to the CUSTOMER_SHIP Object
<ENCRYPTION>	Object / 1	False		Refer to the ENCRYPTION Object
<USERDEFINED>	Object / 1	False		Refer to the USERDEFINED Object
<MPI>	Object / 1	False		Refer to the MPI Object
<SECONDARY_MERCHANT_KEY>	Object / 1	False		Refer to the SECONDARY_MERCHANT_KEY
<DEVELOPERID>	String/8	False		Assigned by SecureNet
<VERSION>	String/10	False		Application version of the client performing the transaction Assigned by SecureNet
<IMAGE>	Object/1	False		Refer to the IMAGE Object
</TRANSACTION>				

5.4 Transaction Image Request

All Gateway transactions include a common set of required parameters (listed first). Additional parameters are required depending on the transaction type. Many optional parameters can also be provided depending on the results that are to be returned.

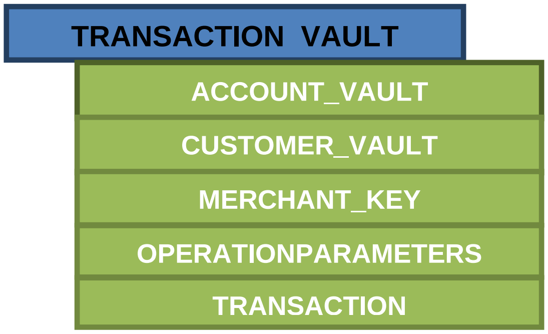
5.4.1 *Transaction Image Request Message Format for Electronic Signature*

XML Tag	Data Type / Max Length	Required	Sample Values	Reference
<IMAGETYPE>	Object / 1	True	ECHECK, SIGNATURE	Available Through: Gateway Services
<MERCHANT_KEY>	Object / 1	True	SECURENET ID SECUREKEY	Refer to MERCHANT_KEY object for additional values
<SIGNATURE>	Byte array	True	Signature in byte array	
<CHECKIMAGE>	Object/ 1	False	Not required	
<TransactionID>	Integer / 15	True	System Generated Unique ID for corresponding transaction, returned by the Gateway	<TransactionID>
<RESPONSE>	Object/ 1	True	Contains response code, sub code and reason text	

6 SECURENET VAULT/TOKENIZATION TRANSACTIONS

SecureNet Vault is SecureNets Payment Card Industry Data Security Standard (PCI-DSS) compliant solution for storing information for credit cards, electronic checks, and Pin-less debit cards. It is a virtual database hosted by and at SecureNet on secure servers. Through tokenization, SecureNet Vault allows tokens to be created for credit card numbers. These tokens are then used for transactions instead of the actual credit card number. Multiple credit card information can be stored for a single individual in the form of tokens. Only the credit card number and expiry date will be stored in SecureNet Vault. The security code cannot be stored as it is against PCI compliance to do so. This means that track data cannot be stored either. In addition, debit cards are excluded, since it is against PCI compliance to store the pin and payments cannot be processed without the pin.

This section outlines the various high level Objects within the SecureNet Vault.

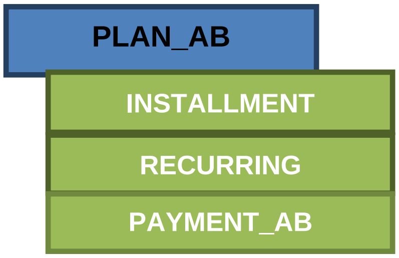


6.1 Vault Specifications

Object Name	Description	Constraints	Methods Used	Reference
ACCOUNT_VAULT	Contains account information for a specific customer within the SecureNet Vault	Account should be set up to use the SecureNet Vault	Methods using this object include: ▪ ProcessAccount ▪ ProcessCustomer ▪ ProcessCustomerAndAccount ▪ ProcessVaultTransaction ▪ CopyVaultAccount	Gateway Services, Extended Transaction Services
CUSTOMER_VAULT	Contains customer information for a specific customer within the SecureNet Vault	Must be enabled to use the SecureNet Vault	Methods using this object include: ▪ ProcessCustomer ▪ ProcessCustomerAndAccount ▪ ProcessVaultTransaction	Gateway Services, Extended Transaction Services
OPERATIONPARAMETERS	This object is used to specify operation type (Add, Update, or Delete)	Must be enabled to use the SecureNet Vault	Methods using this object include: ▪ ProcessVaultTransaction ▪ ProcessAccount ▪ ProcessCustomer ▪ ProcessCustomerAndAccount	Gateway Services,
TRANSACTION_VAULT	Used to process a SecureNet Vault transaction	Must be enabled to use the SecureNet Vault	Methods using this object include: ▪ ProcessAccount ▪ ProcessCustomer ▪ ProcessCustomerAndAccount ▪ ProcessVaultTransaction	Gateway Services,

7 AUTOBILL TRANSACTIONS

This section outlines various high level objects for AutoBill Transactions.

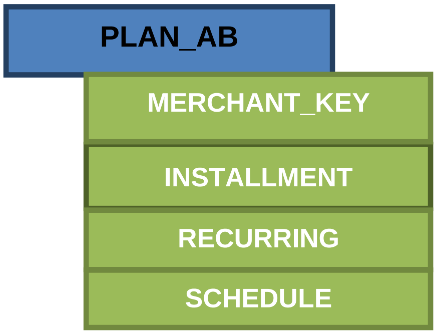


7.1 AutoBill Transaction Specification

Object Name	Description	Constraints	Methods Used	Applicable to
PLAN_AB	This object defines the information that can be submitted to the Gateway for AutoBill transaction processing	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule	Gateway Services, Extended Transaction Services
INSTALLMENT	This object contains information on the customers installment plan	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule	Gateway Services, Extended Transaction Services
RECURRING	This object contains information on the customers recurring payment plan	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule	Gateway Services Transaction Services
PAYMENT_AB	This object contains individual payment information, like amount, payment id, installment date, etc.	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule	Gateway Services Transaction Services

7.1.1 PLAN_AB Object Summary

This section outlines various high level objects within the PLAN_AB Object for AutoBill Transactions.



7.1.2 PLAN_AB Object Specifications

Object Name	Description	Constraints	Methods Used	Applicable to
PLAN_AB	This object defines the information that can be submitted to the Gateway for AutoBill transaction processing	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule ▪ GetABAccount ▪ GetABAccounts	Gateway Services, Extended Services
MERCHANT_KEY	This object is used to verify the merchant before processing any transaction and is required for every operation		▪ All methods	Gateway Services, Extended Services Transaction Services
INSTALLMENT	This object contains information on the customers installment plan	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule ▪ GetABAccount ▪ GetABAccounts	Gateway Services Extended Services
RECURRING	This object contains information on the customers recurring payment plan	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule ▪ GetABAccount ▪ GetABAccounts	Gateway Services Extended Services
SCHEDULE	This object contains individual payment information, like amount, payment id, installment date, etc.	Must be enabled to use AutoBill	Methods using this object include: ▪ AddABAccount ▪ UpdateABAccount ▪ UpdateABSchedule ▪ GetABAccount ▪ GetABAccounts	Gateway Services Extended Services

8 GATEWAY OBJECT PROPERTIES

8.1 MERCHANT_KEY—Required for All Transactions

This object is used to verify merchant before processing any transaction and is required for any operation.

XML Tag	Data Type / Max Length	Required	Description	Reference links
<MERCHANT_KEY>		Required	This object is used to verify the merchant before processing any transaction and is required for every operation	Gateway Services Transaction Services Extended Services
<GROUP_ID>	Integer / 15	False	SecureNet Group ID. Not applicable to all merchants, should be provided by SecureNet Please contact us for more information	
<SECUREKEY>	String / 50	True	Case-sensitive unique SecureKey is associated with corresponding SecureNet ID, used to authenticate the merchant and is required for each submitted transaction	
<SECURENETID>	Integer / 8	True	7-digit unique SecureNet ID provided by SecureNet to identify the merchant and is required for each submitted transaction. If a SecureNet ID is needed—contact SecureNet	
<ADDITIONALINFO>	String / 250	Required if METHOD=EBT	For EBT transactions only. String format: 11 (EBT indicator) followed by a 7-digit FNS number (left-justified, space-filled, if FNS is less than 7 digits)	.
</MERCHANT_KEY>				

8.2 Card

Card payment object contains card information, like card number, expiration date, and can be any of the following: Credit Card, Check Card, Debit Card, or Stored Value Card.

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
<CARD>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<CARDCODE>	String / 4	Required for all E-commerce and MOTO transactions May also be manually keyed-in for Retail transactions for added security and lower interchange rate		A code that is printed on the back of a card. Used as partial assurance that the card is in the buyer's possession. This code is 3 digits long for Visa, MasterCard, and Discover, and 4 digits long for American Express. Not required but could be used for securing credit card transactions by means of card security code validation. If card security code data is submitted, the issuer verifies whether the number matches the number assigned to the card.

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
<CARDNUMBER >	String / 24	Required for Card-Not-Present transactions and manually keyed-in transactions, when TACKDATA is not present. Full card number is required for Transaction code transactions. Last 4 digits of the card number are required for Transaction code transactions. Not required when processing transactions for stored customer records.	Card number, 16 digits long for Visa, MasterCard and Discover, 15 digits long for American Express This value may not contain spaces, non-numeric characters, or dashes	
<EXPDATE>	String / 4	Required for Card-Not-Present transactions and manually keyed-in transactions, when TACKDATA is not present.	A valid card expiration date, which is not in the past Format: MMYY Example: 1012 represents October 2012	
<KSN>	String / 19	Required if METHOD = DB	Debit Only—Unaltered KSN Number from PIN pad	
<PINBLOCK>	String / 16	Required if METHOD = DB	Debit Only—PinBlock from PIN pad	
<TRACKDATA>	String / 200	Required for Card-Present Credit Card transactions, Pin Debit transactions, and EBT (Not Voucher Clear) transactions.	This field contains information encoded from a valid magnetic stripe It includes information such as Primary Account Number and Expiration Date Refer to Appendix for more information.	
<ADDITIONALINFO>	String/ 250	Required if METHOD= EBT and Transaction Type Code=0300 for Voucher Clear Transactions.	These are the possible values for ADDI: 00—Future use. No credit card account verification, default* 01—Future use. Credit card account verification* 11 followed by a 15-character long left-justified space-filled EBT Voucher number ** 02—Updates the EXPDATE of credit card during ProcessTransaction by Customer ID, if expiration date supplied is a future date for the PaymentID	
/CARD>				

*Applicable only when adding a new Account to the SecureNet Vault.

**For EBT Voucher Clear transactions only.

8.3 Check

Check payment object is required for electronic check transactions and contains information on corresponding checking or savings account.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<CHECK>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<ABACODE>	String / 9	True	A 9-digit value representing Bank Routing/ABA Number to identify the Bank.	
<ACCOUNTNAME>	String / 50	True	Account Holder Name is required for Check Verification transactions If left blank—will default to Customer Name from CUSTOMER_BILL object If Customer Name is blank—an error will be returned	
<ACCOUNTNUM>	String / 17	True	Indicated the Account Number to be charged for the transaction, an all-numeric value.	
<ACCOUNTTYPE>	String / 8	True	Type of Bank Account, i.e. SAVINGS, CHECKING,BCHECK, BSAVE	Case sensitive
<BANKNAME>	String / 50	False	Name of Account Holders' Bank.	
<CHECKNUM>	String / 20	Required if SECCODE is BOC, POS, or ARC	Check Number	
<MICRDATA>	String / 400	False	MICR Data Read from Check reader	
<SECCODE>	String / 20	True	ACH Transaction Classification—The Standard Entry Class (SEC) code is a three letter code that identifies the nature of the ACH entry, like: PPD, WEB, TEL, etc. Refer to Appendix	Case sensitive
<ADDITIONALINFO>	String / 250	Used only for METHOD = ECHECK	00—No check verification, default 01—Check verification with Certegy* 02—Pre-Note with ACH Provider**	
</CHECK>				

*Merchant should be enabled with Certegy, please contact SecureNet for more information. For Transaction Type Code 0000-AUTH_ONLY or 0100-AUTH_CAPTURE data will be sent for processing only if check verification returns an APPROVED status from Certegy. If Transaction Type Code is 0700-VERIFICATION data will be sent to Certegy for check verification ONLY, no transaction will be sent for processing.

**Set transaction type code to 0700-VERIFICATION. Data will be sent to ACH Provider for Pre-Noting ONLY, no transaction will be sent for processing. Pre-Note CANNOT be used in conjunction with Transaction Type Codes 0000-AUTH_ONLY or 0100-AUTH_CAPTURE.

8.4 CUSTOMER_BILL

This object contains the customers billing information, like billing address and telephone number, and is required for all E-Commerce transactions.

ML Tag	Data Type / Max Length	Required	Notes	Reference
<CUSTOMER_BILL>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<ADDRESS>	String / 60	True for E-Commerce transactions	Billing street address. Together with Zip Code can be used for securing credit card transactions by means of the Address Verification System (AVS), result of AVS check will be coming back in AVS_RESULT_CODE variable of the TRANSACTIONRESPONSE object.	Alpha number only, no special character
<CITY>	String / 40	True for E-Commerce transactions	Billing City	
<COMPANY>	String/50	False	Billing Company	Alpha number only, no special characters
<COUNTRY>	String / 60	False	Billing Country	Alpha number only, no special characters
<EMAIL>	String / 255	False	Cardholder's Email address Has to be specified if Email receipt is to be sent to the Cardholder for successful transaction	Valid email address
<EMAILRECEIPT>	String/5	False	Merchants can choose to send a Gateway generated email receipt to customers, who provide an email address with their transaction The email receipt includes a summary and results of the transaction. Possible values: TRUE / FALSE If no email is provided value is considered FALSE	Upper case values only
<FIRSTNAME>	String / 50	False	Contains the first name of the customer associated with the billing address for the transaction	
<LASTNAME>	String / 50	False	Contains the last name of the customer associated with the billing address for the transaction	
<PHONE>	String / 25	False	Contains the phone number of the customer associated with the billing address for the transaction	
<STATE>	String / 40	False	Billing State	
<ZIP>	String / 20	True for E-Commerce transactions	Billing Zip Code If full address is not provided, it is advised to provide at least the ZIP for more security	
</CUSTOMER_BILL >				

8.5 CUSTOMER_SHIP

This object contains the customers shipping information, like name and address.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<CUSTOMER_SHIP>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<ADDRESS>	String / 50	False	Customers shipping Street Address	
<CITY>	String / 40	False	Customers shipping City	
<COMPANY>	String / 50	False	Customers shipping Company	
<COUNTRY>	String / 50	False	Customers shipping Country	
<FIRSTNAME>	String / 50	False	Shipping First Name	
<LASTNAME>	String / 50	False	Shipping Last Name	
<STATE>	String / 40	False	Shipping State	
<ZIP>	String / 20	False	Shipping Zip Code	
</CUSTOMER_ SHIP>				

8.6 ENCRYPTION

This object is used for P2PEncryption.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<ENCRYPTION>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<ENCRYPTIONMODE>	Integer	Yes	0—No Encryption 1—P2PE software base 2—P2PE Hardware base	
<KSI>	String	Yes	Integer ID of the P2P Public Key to use for encryption/ decryption. This value represents either the Primary or Secondary active P2P Public Key	
<KEYID>	String	Yes for P2PE software base	Yes	
</ ENCRYPTION >				

8.7 LEVEL2

This Object is used for Level 2 data. Level 2 transactions normally involve corporate cards issued from a U.S Bank.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<LEVEL2>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<DUTY>	Decimal	False*	Duty Amount	
<FREIGHT>	Decimal	False*	Freight Amount	
<PONUM>	String/50	False*	Purchase Order	
<TAX>	Decimal	False*	Tax amount	

XML Tag	Data Type / Max Length	Required	Notes	Reference
<TAXFLAG>	String	False*	0—tax not included 1—tax included 2—tax exempt	
</LEVEL2>				

*Required for Level 2 processing, if this data is not provided—transaction will still be processed as Level 1.

8.8 LEVEL3

This object is used for Level 3 data. Purchases that qualify as Level 3 transactions generally are made with government credit card or corporate cards.

NOTE: *Level 3 is only available to merchants on First Data Nashville platform and only for E-Commerce, Retail, and MOTO transactions.*

XML Tag	Data Type / Max Length	Required	Notes	Reference
<LEVEL3>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<COUNTRYCODETO>	String / 3	False*	Country code shipped to	
<LEVEL3PRODUCTS>	Object Array	False*	Array of LEVEL3PRODUCT	
<MERCHANT_VAT_REGISTRATION_NUM>	String / 20	False*	Merchant VAT Registration Number	
<ORDER_DISCOUNT_AMT>	Decimal / 15	False*	Order Discount Amount	
<PURCHASER_ORDER_DATE>	String / 10	False*	Purchase Order Date	
<PURCHASER_VAT_REGISTRATION_NUM>	String / 20	False*	Purchaser VAT Registration Number	
<SUMMARY_COMMODITY_CODE>	String / 4	False*	Summary Commodity Code	
<VATTAX_RATE>	Decimal / 15	False*	VAT Tax Rate	
<ZIPFROM>	String / 20	False*	Zip Code Shipped From	
<ZIPTO>	String / 20	False*	Zip Code Shipped To	
<ALTTAX_AMT>	Decimal / 15	False*	Alternate tax Amount	
<ALTTAX_AMT_INDICATOR>	String / 1	False*	Alternate tax indicator, Y/N	
<VATTAX_AMT>	Decimal / 15	False*	VAT Tax Amount	
<VAT_INVOICE_REFNUM>	String / 15	False*	VAT invoice Reference Number	
</ LEVEL3>				

*Required for Level 3 processing, if this data is not provided—transaction will still be processed, but will be downgraded to Level 2 (if all Level 2 required data is present) or to Level 1.

8.8.1 LEVEL3PRODUCT

XML Tag	Data Type / Max Length	Required	Notes	Reference
<LEVEL3PRODUCT>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<ALLTAXID>	String / 3	False*		
<COMMODITYCODE>	String/12	False*		
<DISCOUNTAMT>	Decimal	False*		
<DISCOUNTRATE>	Decimal / 15	False*		
<DISCOUNT_INDICATOR>	String / 10	False*		
<GROSSNET_INDICATOR>	String / 20	False*		
<ITEMCODE>	String / 12	False*		
<ITEMNAME>	String / 35	False*		
<ITEMTOTALAMT>	Decimal	False*		
<QUANTITY>	Decimal	False*		
<TAXAMT>	Decimal / 15	False*		
<TAXRATE>	String	False*		
<TAXTYPEIDENTIFIER>	String	False*		
<TAXTYPE_APPLIED>	String / 4	False*		
<UNIT>	String/12			
<UNITPRICE>	Decimal	False*		
</ LEVEL3>				

*Required for Level 3 processing, if this data is not provided—transaction will still be processed, but will be downgraded to Level 2 (if all Level 2 required data is present) or to Level 1.

8.9 MPI

This object describes Merchant Plug-in Indicator for 3D Secure Transactions.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<MPI>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<AUTHINDICATOR>	String / 50	False	The electronic commerce indicator (ECI) value for a Visa transaction; or the universal cardholder authentication field (UCAF) indicator for MasterCard transaction.	
<AUTHVALUE>	String / 50	False	The cardholder authentication verification value (CAVV) for Visa transactions; or accountholder authentication value (AVV)/ universal cardholder authentication field (UCAF) for MasterCard transactions.	
</MPI>				

8.10 Auto

This object is used for Rental industry only and includes rental-related information, like city rented and returned in, rental and return times, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<AUTO>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<EXTRA_CHARGEAMOUNT>	Decimal / 15	True	Additional Charge Amount	
<EXTRA_CHARGECODE>	String / 10	True	Additional Charge Code	
<PREFERRED_CUSTOMER>	String / 1	True	Preferred Customer, Y/N	
<RENTALCITY>	String / 50	True	City Rented in	
<RENTALDATETIME>	String / 14	True	mmddyyyyhhMMss	
<RENTALSTATE>	String / 2	True	State Rented in	
<RETURNCITY>	String / 59	True	City Returned In	
<RETURNDATETIME>	String / 14	True	mmddyyyyhhMMss	
<RETURNSTATE>	String / 2	True	State Returned in	
<SALECODE>	String / 2	False	1—Sale 2—No Show	
</AUTO>				

8.11 Petroleum

This object is used for Petroleum industry only and includes information, like number of gallons, unit price, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<PETROLEUM>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<DRIVEJOBNUM>	String / 6	False	Driver Job Number	
<FLEET_REFERENCENUM>	String / 10	False	Fleet Reference Number	
<FULLSERV_INDICATOR>	String / 3	False	Full Service indicator	
<GALLONS>	Integer / 15	False	Number of Gallons	
<ODOMETER>	String / 6	False	Odometer reading	
<PIN>	String / 10	False		
<PRODUCT>	Array of PRODUCT Objects	True	Product Listing	Refer to Appendix A for Petroleum Product Codes.
<TAXAMOUNT>	Decimal / 15	False	Tax Amount	
<UNITPRICE>	Decimal / 15	False	Price per unit	
<VEHICLENUM>	String / 6	False	Vehicle Number	
</PETROLEUM>				

8.12 Hotel

This object is used for Hotel industry only and includes information, like check-in, check-out times, room rate, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<HOTEL>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<CHARGETYPE>	String / 1	True	1—Hotel /Lodging 2—Restaurant 3—Gift Shop	
<CHECKIN_DATE>	String / 14	True	mmddyyyy	
<CHECKOUT_DATE>	String / 14	True	mmddyyyy	
<EXTRA_CHARGEAMOUNT>	Decimal / 15	True	Additional Charge Amount	
<EXTRA_CHARGECODE>	String / 10	True	Additional Charge Code	
<PREFERRED_CUSTOMER>	String / 1	True	Preferred Customer, Y/N	
<ROOMRATE>	Decimal / 15	True	Rate of room	
<SALECODE>	String / 2	True	1—Sale 2—No Show 3—Deposit 4—Delayed Charge 5—Express Service 6—Assured Reservation	
</HOTEL>				

8.13 Product

This object contains individual product information, like product description, amount, quantity, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<PRODUCT>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<AMOUNT>	Decimal / 15	False	Product Amount	
<CODE>	String / 9	False	Product Code	
<DESCRIPTION>	String / 50	False	Product Description	
<NAME>	String / 30	False	Product Name	
<QUANTITY>	String / 10	False	Product Quantity	
<TAX>	Decimal / 15	False	Product Tax Amount	
<TAXABLE>	String / 3	False	0 = No, 1 = Yes	
<UNIT>	String / 9	False	Unit of Measure	
<UNITPRICE>	Decimal / 15	False	Price per Unit	
</PRODUCT>				

8.14 Service (Restaurant)

This object is used for the Restaurant industry and contains gratuity amount and server number.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<SERVICE>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<GRATUITY>	Decimal / 15	False	Gratuity Amount	
<SERVERNUM>	String / 6	False	Server Number	
</SERVICE>				

8.15 UserDefined

This object contains 50 user defined fields and allows merchants to store additional data along with transaction information. Please note that user defined fields are not intended to and must not be used to capture personally identifying information, like name, address, credit card number, social security number, driver's license number, state-issued identification number, passport number, and card verification numbers.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<USERDEFINED>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<UD1>	String / 50	False	Additional Data Field	
<UD2>	String / 50	False	Additional Data Field	
….	String / 50	False	Additional Data Field	
<UD50>	String / 50	False	Additional Data Field	
</USERDEFINED>				

8.16 SECONDARY_MERCHANT_KEY

This object is similar to the MERCHANT_KEY Object and is used for processing transactions across various SecureNet IDs using information from the Vault linked to the Primary SecureNet ID.

XML Tag	Data Type / Max Length	Required	Description	Reference
<SECONDARY_MERCHANT_KEY>			This object is used to verify the merchant before processing any transaction and is required for every operation	Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<GROUP_ID>	Integer / 15	False	SecureNet Group ID—Not applicable to all merchants, should be provided by SecureNet Please contact SecureNet for more information.	
<SECUREKEY>	String / 50	True	Case-sensitive unique SecureKey is associated with corresponding SecureNet ID, used to authenticate the merchant and is required for each submitted transaction.	

XML Tag	Data Type / Max Length	Required	Description	Reference
<SECURENETID>	Integer / 8	True	7-digit unique SecureNet ID provided by SecureNet to identify the merchant and is required for each submitted transaction If a SecureNet ID is needed—contact SecureNet	
<ADDITIONALINFO>	String / 250	Required if METHOD=EBT	For EBT transactions only. String format: 11 (EBT indicator) followed by a 7-digit FNS number (left-justified, space-filled, if FNS is less than 7 digits).	.
</SECONDARY_MERCHANT_KEY>				

8.17 Terminal

This object includes merchant-related information on locations and terminal. If not provided, this information will be taken from the merchants records. If provided, this object will overwrite any existing data in the merchants records.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<TERMINAL>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<STORENUMBER>	String / 10	Required for Ceridian		
<TERMINALCITY>	String / 27	Required if SECCODE POS AND Several locations are being processed via one single SecureNet ID	Terminal City—This field identifies the city, town, village, or township in which an Electronic terminal is located	
<TERMINALID>	String / 20	Required if SECCODE POS AND Several locations are being processed via one single SecureNet ID	Terminal Identification Code—This filed identifies an Electronic terminal with a unique code that allows a terminal owner and/or switching network to identify the terminal at which an Entry originated	
<TERMINALLOCATION>	String / 27	Required if SECCODE POS AND Several locations are being processed via one single SecureNet ID	Terminal Location—This field identifies the specific location of a terminal (i.e. street names of an intersection, address, etc.) in accordance with the requirements on Regulation E. Section 205.9(a)(5) requires that an ATM or POS terminal receipt contain the location of the terminal where the transfer is initiated, or an identification such as a code or terminal number. Comment 9(a)(5)–1 is revised, as proposed, to clarify that either a code or location may be disclosed. Comments 9(a) (5) (iv)–1 and –2 are re-designated as comments 9(a) (5)–3 and –4.	

XML Tag	Data Type / Max Length	Required	Notes	Reference
<TERMINALSTATE>	String / 10	Required if SECCODE POS AND Several locations are being processed via one single SecureNet ID	Terminal State—This field identifies state of the United States in which an Electronic terminal is located	
</TERMINAL>				

8.18 Healthcare

XML Tag	Data Type / Max Length	Required	Notes	Reference
<HEALTHCARE>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<SERVICE_TYPECODE>	String / 2	Required		
<CARRIERID>	String / 6	Required	This is Insurance company Id (Provider Id)	
<OTC_AMOUNT>	decimal	Required	Over the counter amount	
<PRESCRIPTION_AMOUNT>	decimal		Prescription Amount	
<VISION_AMOUNT>	decimal		Vision Amount	
<CLINIC_AMOUNT>	decimal		Clinic Amount	
<DENTAL_AMOUNT>	decimal		Dental Amount	
</HEALTHCARE>				

8.19 Image

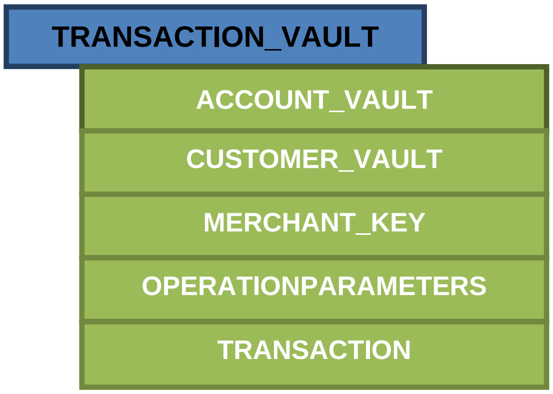
XML Tag	Data Type / Max Length	Required	Notes	Reference
<IMAGE>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<SIGNATURE >	Byte Array	Required for Signature update		
<CHECK IMAGE>	Object/1	Required for Check21 transactions for check image capturing		Refer to the CHECKIMAGE Object
</IMAGE>				

8.19.1 Checkimage

XML Tag	Data Type / Max Length	Required	Notes	Reference
<CHECKIMAGE>				Refer to Section 4.1 and 4.2 for methods and section 7.1 for Transaction Object Specification.
<FRONTSIDE>	Byte Array	Required		
<BACKSIDE>	Byte Array			
</CHECKIMAGE>				

8.20 Transaction Vault Object Summary

This object defines information that can be submitted to the Gateway for Vault transaction processing.



8.21 Transaction Vault Object Specification

8.21.1.1 TRANSACTION_VAULT

This object defines information that can be submitted to the Gateway for Vault transactions processing.

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
			Methods using this object include: ▪ ProcessAccount ▪ ProcessCustomer ▪ ProcessVaultTransaction ▪ ProcessCustomerAndAccount	
<TRANSACTION_VAULT>				Refer to Section 4.1 and 4.2 for methods and section 8.2 for Vault Object Specification.
<ACCOUNT_VAULT>	Object / 1	True		Refer to the ACCOUNT_VAULT Object
<CUSTOMER_VAULT>	Object / 1	True		Refer to the CUSTOMER_VAULT
<MERCHANT_KEY>	Object / 1	True		Refer to the MERCHANT_KEY Object
<OPERATIONPARAMETERS>	Object / 1	True		Refer to the OPERATIONPARAMETERS Object
<TRANSACTION>	Object / 1	True		Refer to the TRANSACTION Object
</TRANSACTION_VAULT>				

8.21.2 ACCOUNT_VAULT

This object contains account information to add, update or delete, like payment method and corresponding card/account information.

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
			Methods using this object include: ▪ ProcessAccount ▪ ProcessCustomer ▪ ProcessVaultTransaction ▪ ProcessCustomerAndAccount ▪ CopyVaultAccount	
<ACCOUNT_VAULT>				Refer to Section 4.1 and 4.2 for methods and section 8.2 for Vault Object Specification.
<ACDI>	Integer / 1	False	Account Duplicate Indicator. Values: 0, 1, 2, 3. Refer to ACDI in Appendix A for more information.	
<CARD>	OBJECT / 1	Required if METHOD = CC, PD		Refer to the CARD Object
<CHECK>	OBJECT / 1	Required if METHOD = ECHECK		Refer to the CHECK Object
<CUSTOMERID>	String / 25	True	Contains customer ID associated with transaction or record. Can be any string or AUTO. If AUTO is specified a unique CUSTOMERID is generated and returned in the response object. It cannot be a valid card number.	Recommended to use specific customer id than AUTO. AUTO is case sensitive
<CUSTOMER_BILL>	OBJECT / 1	Required for E-Commerce transactions		Refer to CUSTOMER_BILL Object.
<METHOD>	String / 7	True	Indicates method of payment for the transaction being sent to the system.	Refer to Appendix A for more information.
<NOTES>	String / 250	False	Additional Notes field.	
<PAYMENTID>	String / 25	True	Contains the Payment ID associated with the customer record. Can be any string or AUTO. If AUTO is specified, a unique PAYMENTID is generated and returned in the response object. It cannot be a valid card number.	Recommended to use specific payment id than AUTO. AUTO is case sensitive
<PRIMARY>	String / 5	False	Specifies if the account is Primary or not. Values: TRUE or FALSE. Set to FALSE by default. Multiple accounts (payment methods) can be stored for a single customer and only one account can be set as primary. The system will automatically default to another payment method for the customer if the primary payment is declined.	Upper case values only

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
<USERDEFINED>	OBJECT / 1	False		Refer to the USERDEFINED Object
<CARDTYPE>	STRING			Returned in the response
<LASTACCESSDATE>	STRING			Returned in the response
</ACCOUNT_VAULT>				

8.21.3 CUSTOMER_VAULT

This object contains customer information to add, update or delete, like customer name, address, email, phone number, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference	XML Tag
<CUSTOMER_VAULT>				Refer to Section 4.1 and 4.2 for methods and section 8.2 for Vault Object Specification	<CUSTOMER_VAULT >
<CSDI>	Integer / 1	True	Customer Duplicate Check Indicator 0—If Customer ID exists—return an Error 1—If Customer ID exists—do not add and continue with transaction.		<CSDI>
<CUSTOMERID>	String / 25	True	Contains the customer ID associated with the transaction or record. Can be any string or AUTO. If AUTO is specified a unique CUSTOMERID is generated and returned in the response object. It cannot be a valid card number.		<CUSTOMERID> Recommended to use specific customer id than AUTO. AUTO is case sensitive
<CUSTOMER_BILL>	Object / 1	False	Object containing Billing Customer information.		<CUSTOMER_BILL>
<NOTES>	String / 200	False	Additional Notes field.		<NOTES>
<USERDEFINED>	Object / 1	False	Object Containing 50 user defined fields.		<USERDEFINED>
</CUSTOMER_VAULT>					</CUSTOMER_VAULT >

8.21.4 OPERATIONPARAMETERS

This object is used to specify operation type (Add, Update, or Delete) and to specify whether Vault information is to be added even when transaction is declined in *ProcessVaultTransaction* method.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<OPERATIONPARAMETERS>				Refer to Section 4.1 and 4.2 for methods and section 8.2 for Vault Object Specification.
<ACTIONCODE>	Integer / 1	False	1—ADD 2—Update 3—Delete	
<ADD_IF_DECLINED>	Integer / 1	False	Add Customer/Account if transaction is declined: 0—False 1—True	
</OPERATIONPARAMETERS>				

8.22 AutoBill Transactions

The AutoBill feature allows merchants to set up automatic scheduled and recurring credit card and ACH transactions. Payments can be scheduled for automatic daily, weekly, monthly, quarterly, and yearly recurrence, as well as any interval in between. In addition, merchants can set up one or more single scheduled future payments for any date. The automatic installment generator can take a total amount due and evenly split the payment dates and payment amounts across a designated time frame.

8.22.1 PLAN_AB Object Specification

8.22.1.1 PLAN_AB

This object defines the information that can be submitted to the Gateway for AutoBill transaction processing.

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
<PLAN_AB>			This object can be used with the following methods: ▪ AddABAAccount ▪ UpdateABAccount ▪ UpdateABSchedule	Available through:
<ACTIVE>	Integer	False	Available in response. Shows whether recurring billing on this record is enabled or disabled. 0 – Inactive, 1—Active	
<CUSTOMERID>	String / 25	True	Contains customer ID associated with payment plan. It cannot be a valid card number.	
<ENDDATE>	String / 8	Required if NOEND_FLAG = 1 (recurring bill has an end date)	Plan end date. MMDDYYYY.	
<INSTALLMENT>	Object / 1	True		Refer to PLAN_AB Object Summary

XML Tag	Data Type / Max Length	Required	Methods Used	Reference
<MAXRETRIES>	Integer / 1	False	AutoBill offers the ability to reattempt a declined transaction every day for up to five consecutive days. Enter maximum number of retries, default is 0, maximum is 5.	
<MERCHANT_KEY>	Object / 1	True	This object is used to verify merchant before processing any transaction and is required for any operation.	Refer to MERCHANT_KEY Object
<NEXTPAYMENT_DATE>	String / 8	False	Available in response. Next payment date. MMDDYYYY.	
<PAYMENTID>	String / 25	True	SecureNet Vault Payment ID. Must be enabled on the account. It cannot be a valid card number.	
<PAYMENTID2>	String / 25	False	Process through Payment ID 2 if there is a problem with Payment ID.	
<PLANID>	Integer	Required for Update only	Payment plan ID, auto-generated by the system and given to the merchant at the time Plan is added.	
<RECURRING>	Object / 1	True		Refer to RECURRING Object
<SCHEDULE>	Array of PAYMENT_AB	True		Refer to PAYMENT_AB Object
<STARTDATE>	String / 8	True	Plan start date. MMDDYYYY. Verify the date is on or after current date.	
<TYPE>	String / 3	True	Billing types offered include Recurring Billing, Installment Billing and Variable Billing. Set TYPE to one of the following: REC—for recurring INS—for installment VAR—for variable	
</PLAN_AB>				

8.22.1.2 *OPTIONS*

Please note that OPTIONS is a parent class of INSTALLMENT and RECURRING, so all parameters of OPTIONS are inherited by the two subclasses and are included in INSTALLMENT and RECURRING tables below.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<OPTIONS>				
<CYCLE>	String / 1	True	W—weekly M—monthly Q—quarterly B—semiannual Y—yearly	

XML Tag	Data Type / Max Length	Required	Notes	Reference
<DAY>	Integer / 2	Required if CYCLE=M, Q, B, or Y	1-28	
<FREQUENCY>	Integer / 2	True	Once every week/month/ quarter/six months/year, as following: If CYCLE=M: 1-11 if CYCLE=W: 1-51 if CYCLE=Q: 1-3 for CYCLE=Y or CYCLE=B—frequency is set automatically	
<MONTH>	Integer / 2	Required if CYCLE=Q, B, or Y	1-12	
<WEEKDAY>	Integer / 1	Required if CYCLE=W	1-7, where 1 is Sunday	
</OPTIONS>				

8.22.1.3 INSTALLMENT

Installment payment plans consist of a single purchase of goods or services that is billed to an account in multiple segments, over a period of time, with payments occurring on a schedule agreed by a cardholder and merchant. The INSTALLMENT object below contains information on the Customers Installment plan.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<INSTALLMENT>				
<CYCLE>	String / 1	True	W—weekly M—monthly Q—quarterly B—semiannual Y—yearly	
<DAY>	Integer / 2	Required if CYCLE=M, Q, B, or Y	1-28	
<FREQUENCY>	Integer / 2	True	Once every week/month/quarter/six months/year, as following: If CYCLE=M: 1-11 if CYCLE=W: 1-51 if CYCLE=Q: 1-3 for CYCLE=Y or CYCLE=B—frequency is set automatically	
<MONTH>	Integer / 2	Required if CYCLE=Q, B, or Y	1-12	
<WEEKDAY>	Integer / 1	Required if CYCLE=W	1-7, where 1 is Sunday	
<AMOUNT>	Decimal / 15	True	Installment amount	
<AUTOCALC_OPTION>	String / 1	True	A—automatic M—manual	
<BALLOON_AMOUNT>	Decimal / 15	True	Balloon amount	

XML Tag	Data Type / Max Length	Required	Notes	Reference
<BALLOON_OPTION>	Integer / 1	True	AutoBill offers the capability of the balloon payment to be applied to the first or last payment, as following: 0—add balloon amount to first payment 1—add balloon amount to last payment	
<COUNT>	Integer / 2	True	Number of total installments for installment plan. 1-99.	
<REMAINDER_OPTION>	Integer / 1	True	AutoBill offers the ability to choose first or last payment if installments are not equally divided, as following: 0—add remainder to first payment 1—add remainder to last payment	
</INSTALLMENT>				

8.22.1.4 RECURRING

Recurring payment plans exist when multiple transactions are processed at pre-determined intervals, as a result of an agreement for the purchase of products or services that are provided over time. Recurring payment plans help simplify the process of billing a cardholder for a product or a service that is being provided on a continuous basis.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<RECURRING>				
<CYCLE>	String / 1	True	W—weekly M—monthly Q—quarterly B—semiannual Y—yearly	
<DAY>	Integer / 2	Required if CYCLE=M, Q, B, or Y	1-28	
<FREQUENCY>	Integer / 2	True	Once every week/month/quarter/six months/year, as following: If CYCLE=M: 1-11 if CYCLE=W: 1-51 if CYCLE=Q: 1-3 for CYCLE=Y or CYCLE=B—frequency is set automatically	
<MONTH>	Integer / 2	Required if CYCLE=Q, B, or Y	1-12	
<WEEKDAY>	Integer / 1	Required if CYCLE=W	1-7, where 1 is Sunday	
<AMOUNT>	Decimal / 15	True	Total value to be charged to the payment method specified. Include decimal point followed by decimal amount.	
<NOEND_FLAG>	Integer / 1	True	0—Recurring bill has no end date 1—Recurring bill has an end date	
</RECURRING>				

8.22.1.5 PAYMENT_AB

This object contains individual payment information, like amount, payment id, installment date, etc.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<PAYMENT_AB>				
<AMOUNT>	Decimal / 15	True	Total amount to be charged to the payment method specified. Include decimal point followed by decimal amount.	
<INSTALLMENTDATE>	String / 8	True	Installment date. MMDDYYYY.	
<INSTALLMENTNUM>	Integer / 2	True	Number of installments. 1-99.	
<NUMOFRETRIES>	Integer / 1	True	AutoBill offers the ability to reattempt a declined transaction every day for up to five consecutive days. Enter number of retries, between 1-5.	
<PAID>	Integer	False	Available in response. Will contain an array of all records for Installment and Variable billing, so that the merchant can see which has already been paid or not.	
<PAYMENTDATE>	String	False	Available in response. Will contain payment date of that Schedule ID for Installment and Variable billing, for Recurring billing—represents next billing date.	
<PAYMENTID>	String / 25	True	Contains the Payment ID associated with the customer record. Can be any string or AUTO. If AUTO is specified, a unique PAYMENTID is generated and returned in the response object.	
<PLANID>	Integer	Required for Update only	Payment plan ID, auto-generated by the system and given to the merchant at the time Plan is added.	
<PROCESSED>	Integer / 1	False	0—not yet processed, still in the queue 1—processed	
<SCHEDULEID>	Integer	Required for Update only	AutoBill allows multiple billing schedules per Customer ID. Schedule ID, auto-generated by the system and given to the merchant at the time Schedule is added.	
<TransactionID>	Integer / 15	False	System Generated Unique ID for corresponding transaction, returned by the Gateway.	
</PAYMENT_AB>				

8.23 Settlement Transactions

In order to close a batch the below object is required.

Object Name	Description	Reference
MERCHANT_KEY	Required for CloseBatch method.	

8.24 Reporting API Services

Reporting API Services allow merchants to perform additional functions with their processing systems. The following services are available through the Reporting API:

> ***NOTE:*** *Data in the reporting services database is not updated in real-time, there is a 30-minute delay.*

- Transactions Services:
  - Retrieve Transactions Info by Batch ID
  - Retrieve Current Transaction Info by Order ID
  - Retrieve Current Transaction Info by Transaction ID
- Vault Services:
  - Allow to lookup Customer and Account information
- AutoBill Services:
  - Allow to lookup AutoBill Account information
- Batch Services:
  - Allow the system to initiate a batch close on demand
- P2P Encryption Services:
  - Request a new P2P Encryption Key
  - Confirm a new P2P Encryption Key
  - Retrieve the current Active Primary or Active Secondary P2P Encryption Key

To implement any of the above functions—create a program to consume one of the following Gateway Services:

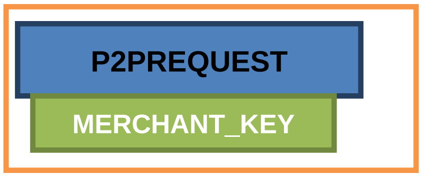
- For Batch and Transaction Services:
  - URL for Production: https://gateway.securenet.com/api/data/transaction.svc
  - URL for Certification and Test: https://certify.securenet.com/api/data/transaction.svc
- For SecureNet Vault and AutoBill Services:
  - URL for Production: https://gateway.securenet.com/api/data/service.svc
  - URL for Certification and Test: https://certify.securenet.com/api/data/service.svc
- For P2P Encryption Services:
  - URL for Production: https://gateway.securenet.com/api/data/service.svc
  - URL for Certification and Test: https://certify.securenet.com/api/data/service.svc

Request Objects used for Reporting API Services have been described above and are the following:

Object Name	Description	Reference
MERCHANT_KEY	Required for all Reporting API methods except GetABAccount.	
P2PRequest	Required for requesting a new P2P Encryption Key	
P2PConfirmationRequest	Required for Confirmation and Activation of a new P2P Key.	

8.25 P2PREQUEST Object

This section outlines the various sections within the P2PRequest object.



P2PREQUEST	This object defines the fields required for requesting and confirming a new P2P Encryption Key.
MERCHANT_KEY	A MERCHANT_KEY representing the current client.

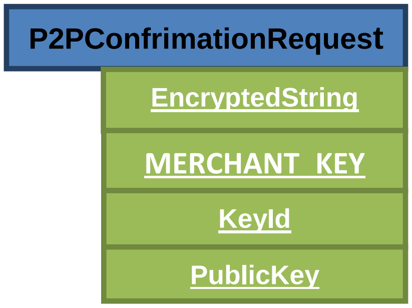
8.25.1 *P2PREQUEST Object Specification*

This object defines information that can be submitted to the P2P Encryption service for processing when requesting a new P2P Encryption Key.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<P2PREQUEST>				
<MERCHANT_KEY>	Object / 1	True		Refer to MERCHANT_KEY Object
</P2PREQUEST>				

8.25.2 *P2PConfirmationRequest Object Structure*

This section outlines the various sections within the P2PConfirmationRequest Object.



8.25.3 *P2PConfirmationRequest Object Specification*

This object defines information that can be submitted to the P2P Encryption service for processing when confirming and activating a new P2P Key.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<P2PConfirmationRequest >				Available through:
<ENCRYPTED_STRING>	String/172	True	Gets or sets the encrypted string used during confirmation of the P2P Key.	Extended Services
<KEYID>	Integer	True	Gets or sets the ID of the P2P Key.	Extended Services
<MERCHANT_KEY>	Object	True	Gets or sets the requesting client's MERCHANT_KEY. (Inherited from P2PRequest.)	
</P2PConfirmationRequest>				

8.25.4 P2PPublicKeyInfo

This object defines the P2P Encryption Public Key information that is used by the client.

XML Tag	Data Type / Max Length	Notes	Reference
<P2PPublicKeyInfo>			Available through:
<Exponent>	String / 4	The Exponent variable used for calculating the encryption key.	Extended Services
<Modulus>	String / 172	The Modulus variable used for calculating the encryption key.	Extended Services
</ P2PPublicKeyInfo >			

9 RESPONSES

This section describes the responses returned by the Gateway when a merchant server submits a transaction for processing. These responses are objects that contain 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.

9.1 Standard Transaction Responses Structure

This section describes the response returned by the Gateway when a merchant server submits a transaction for processing.

GATEWAYRESPONSE
ABRESPONSE
SCHEDULE
TRANSACTIONRESPONSE
CUSTOMER_BILL
VAULTACCOUNTRESPONSE
VAULTCUSTOMERRESPONSE
P2PKEYRESPONSE
GATEWAYRESPONSE

9.1.1 GATEWAYRESPONSE Object Specification

Property Name	Type	Description	Constraints	Reference
ABRESPONSE	Object	Response for AutoBill Transactions Can be part of GATEWAYRESPONSE or standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Gateway Services
TRANSACTIONRESPONSE	Object	Response for Standard Payment Transactions and Reporting API Transactions Can be part of GATEWAYRESPONSE or standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Gateway Services Transaction Services
VAULTACCOUNTRESPONSE	Object	Response for SecureNet Vault Account Transactions Can be part of GATEWAYRESPONSE or Standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Gateway Services
VAULTCUSTOMERRESPONSE	Object	Response for SecureNet Vault Customer Transactions Can be part of GATEWAYRESPONSE or standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Gateway Services
ABRESPONSE	Object	Response for AutoBill Transactions Can be part of GATEWAYRESPONSE or standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Gateway Services
P2PKEYRESPONSE	Object	Contains the results of the P2P Encryption request Can be part of GATEWAYRESPONSE or standalone for transactional responses	All restrictions as defined in the Request Object Specs apply.	Extended Services Transaction Services

9.1.1.1 ABRESPONSE

This object defines the information that will be sent by the Gateway as a response for AutoBill transactions.

***NOTE:** All fields in the ABRESPONSE object are returned in a left-justified, space-filled format.*

XML Tag	Data Type / Max Length	Notes	Reference
<ABRESPONSE>			Available through:
<RESPONSE_CODE>	String / 10	Transaction Response Code.	Refer to response codes in the Appendix.
<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.	
<CUSTOMERID>	String / 25	Contains the customer ID associated with the transaction or record.	
<PAYMENTID>	String / 25	Contains the Payment ID associated with the customer record.	
<PLANID>	Integer	Echoed from PLAN_AB	
<SCHEDULE>	Object / 1	Echoed from PLAN_AB	
</ABRESPONSE>			

9.1.1.2 TRANSACTIONRESPONSE

This object defines the information that will be sent by the Gateway as a response for all standard payment transactions, as well as transactions submitted through Reporting API Services.

***NOTE:** All fields in the TRANSACTIONRESPONSE object are returned in a left-justified, space-filled format.*

XML Tag	Data Type / Max Length	Notes	Reference
<TRANSACTIONRESPONSE>			Available through: Gateway Services
<ADDITIONALAMOUNT>	Decimal / 15	For Debit cards it will contain the surcharge for the transaction, if one has been setup on the Host. (If surcharge applies, it must be printed on the customer receipt). Applicable to Paymentech only. For EBT and Stored Value cards it will contain the remaining balance on the card. See Appendix for Additional Data Chart.	

XML Tag	Data Type / Max Length	Notes	Reference
<ADDITIONALDATA1>	String / 100	For Stored Value cards. Includes Previous Balance (15 characters long zero filled decimal number with explicit decimal point) followed by Cash Out Amount (15 characters long zero filled decimal number with explicit decimal point). See Appendix for Additional Data Chart.	
<ADDITIONALDATA2>	String / 100	Paymentech specific data -for Debit Transactions it will contain the Trace Number for the transaction. Optional data returned by the Host for Fleet Cards. (If data exists, it must be printed on the customer receipt as reference data). For PD transactions, settlement date is returned. For EBT transactions it will contain the FNS # echoed from the ADDITIONALINFO field of the MERCHANT_KEY object. See Appendix Additional Data Chart.	
<ADDITIONALDATA3>	String / 100	For EBT transactions it will contain the Voucher # echoed from the ADDITIONALINFO field of the CARD object. See Appendix for Additional Data Chart.	
<ADDITIONALDATA4>	String / 100	for ISSUE_VIRTUAL: Virtual issued card Pin Number	
<ADDITIONALDATA5>	String / 100	P2P Encryption Transaction Response in the format: MMddyyyyHHmmss	
<AUTHCODE>	String / 6	The six-digit alphanumeric authorization or approval code.	
<AUTHORIZEDAMOUNT>	Decimal / 15	Authorized Transaction Amount.	
<AVS_RESULT_CODE>	String / 2	Address Verification System Result Code. See Appendix	
<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. See Appendix	
<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. See Appendix	
<CARDTYPE>	String / 10	Card Type Code is returned. See Appendix	
<CASHBACK_AMOUNT>	Decimal / 15	Cash Back Amount echoed from TRANSACTION	
<CAVV_RESPONSE_CODE>	String / 2	Indicates the results of Cardholder Authentication Verification Value (CAVV) verification. See Appendix	
<CHECKNUM>	String / 20	Echoed from TRANSACTION	
<CODE>	String / 4	Echoed from TRANSACTION	
<CUSTOMERID>	String / 25	Echoed from TRANSACTION	
<EXPIRYDATE>	String / 4	Echoed from TRANSACTION	
<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 / 5	Is Level 2 valid	

XML Tag	Data Type / Max Length	Notes	Reference
<LEVEL3_VALID>	String / 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 Authorization Network.	
<OrderID>	String / 25	Echoed from TRANSACTION	
<PAYMENTID>	String / 25	Echoed from TRANSACTION	
<RESPONSE_CODE>	String / 3	Transaction Response Code. See Appendix	
<RESPONSE_REASON_CODE>	String / 6	Code for providing more details about the result of the transaction.	
<RESPONSE_REASON_TEXT>	String / 200	Brief description of the result, which corresponds to the Response Reason Code.	
<RESPONSE_SUBCODE>	String / 6	Code used by the system for internal transaction tracking.	
<RETREFERENCENUM>	String / 50	Bank authorization reference #	
<SECURENETID>	Integer / 8	Echoed from TRANSACTION	
<SETTLEMENTAMOUNT>	Decimal / 15	Settlement amount of Transaction	
<SETTLEMENTDATETIME>	String / 14	Settlement date of transaction (MMDDYYYYHHMMSS).	
<SYSTEM_TRACENUM>	String / 50	System Trace # for Authorization Network	
<TRACKTYPE>	String / 1	Track Type used for transaction, 0 if no track data is present	
<TRANSACTIONAMOUNT>	Decimal / 15	Echoed from TRANSACTION. For 0906-INQUIRY_BALANCE transactions this field indicates the available balance on the gift card.	
<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 / 1	Echoed from TRANSACTION	
</ TRANSACTIONRESPONSE>			

9.1.1.3 *VAULTACCOUNTRESPONSE*

This object defines the information that will be sent by the Gateway as a response for all SecureNet Vault Account transactions.

NOTE: *All fields in the VAULTACCOUNTRESPONSE object are returned in a left-justified, space-filled format.*

XML Tag	Data Type / Max Length	Notes	Reference
<VAULTACCOUNTRESPONSE>	Object / 1	Response for SecureNet Vault Account Transactions.	
<RESPONSE_CODE>	String / 10	Transaction Response Code. See Appendix.	
<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.	

XML Tag	Data Type / Max Length	Notes	Reference
<ACCOUNTTYPE>	String / 8	Echoed from ACCOUNT_VAULT	
<ADDITIONALDATA1>	String / 100	Contains TransactionID, which is generated while verifying an account before adding/updating Vault account information.	
<ADDITIONALDATA2>	String / 100	Future Use	
<CARDTYPE>	String / 10	Card Type Code is returned. See Appendix	
<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	
<TRANSACTIONDATETIME>	String / 14	Transaction date and time as recorded in the Gateway according to merchant time zone. (MMDDYYYYHHMMSS).	
</VAULTACCOUNTRESPONSE>			

9.1.1.4 *VAULTCUSTOMERRESPONSE*

XML Tag	Data Type / Max Length	Notes	Reference
<VAULTCUSTOMERRESPONSE>	Object / 1	Response for SecureNet Vault Account Transactions.	
<RESPONSE_CODE>	String / 10	Transaction Response Code. See Appendix.	
<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.	
<COMPANY>	String		
<CUSTOMERID>	String		
<FIRSTNAME>	String		
<LASTNAME>	String	Echoed from ACCOUNT_VAULT	
<TRANSACTIONDATETIME>	String	Transaction date and time as recorded in the Gateway according to merchant time zone. (MMDDYYYYHHMMSS).	
</VAULTCUSTOMERRESPONSE >			

9.1.1.5 *P2PRESPONSE*

This object defines the information that will be sent by the Gateway as a response when requesting a new P2P Encryption Key or the Current Active Primary or Secondary P2P Encryption Key.

NOTE: *All fields in the P2PNewResponse object are returned in a left-justified format.*

XML Tag	Data Type / Max Length	Required	Notes	Reference
<P2PRESPONSE>	Object			
<COMFIRMCODE>	String/16	True	The confirmation string of a P2P Encryption key.	
<CONFIRMATIONNUMBER>	String/10	True	The confirmation number of a P2P Encryption key.	
<EXPDATE>	String/8	True	The date that the P2P Encryption key is set to expire.	
<KEY>	Object	True	The public key portion of a P2P Encryption key.	
<KEYID>	Integer	True	The unique identifier of the P2P Encryption key.	
<RESPONSE_CODE>	String/1	True	Indicates whether the operation was Approved, Declined, or an Error.	
<RESPONSE_REASON_CODE>	String / 4	True	The unique identifier of the RESPONSE_REASON_TEXT	
<RESPONSE_REASON_TEXT>	String /Variable	True	The human-readable text of the operation response.	
<RESPONSE_SUBCODE>	String / 4	False	The sub-code that categorizes a RESPONSE_CODE	
<STATUS>	Integer	True	The current status of the P2P Encryption key.	
</ P2PRESPONSE >				

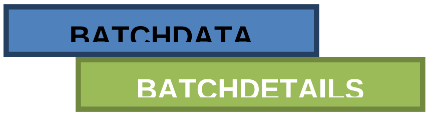
9.1.1.6 *IMAGERESPONSE*

This object defines the information that will be sent by the Gateway as a response when requesting to update the transaction.

XML Tag	Data Type / Max Length	Required	Notes	Reference
<TransactionID>	Integer / 15	True	System Generated Unique ID for corresponding transaction, returned by the Gateway.	
<IMAGEFILENAME >	String	True	List of Image file names.	

9.2 Response Objects for Settlement Transactions

9.2.1 *BATCHDATA Object Structure*



9.2.1 *BATCHDATA Object Specification*

This section details high-level BATCHDATA Object specification. Descriptions, rules and notes for each object of BATCHDATA are detailed below:

XML Tag	Data Type / Max Length	Notes	Reference
<BATCHDATA>			
<BATCHDETAILS>	Array of BATCHDETAIL	Refer to BATCHDETAIL Object.	
</BATCHDATA>			

9.2.1.1 BATCHDETAIL

This object defines the information that can be submitted to the Gateway for Settlement Transactions.

XML Tag	Data Type / Max Length	Notes	References
<BATCHDETAIL>			
<BatchId>	String / 20	Batch ID created on Batch Close Should be the value returned by the Gateway	
<METHOD>	String	CC, DB, ECHECK	
<PROCESSOR>	String	1 –TSYS 2- Paymentech 3- SecureNet Frontend 5- Nashville	
<SECURENETID>	String / 20	7-digit unique SecureNet ID provided by SecureNet to identify the merchant and is required for each submitted transaction If a SecureNet ID is needed—contact SecureNet	
<TOTALBATCHAMOUNT>	String / 20	Total Amount of Sale and Credits	
<TOTALCREDITAMOUNT>	String / 20	Total Amount of Credit	
<TOTALSALEAMOUNT>	String / 20	Total Amount of Sale	
<TOTALTRANSACTIONS>	String / 20	Number of transactions	
<BATCHDETAIL>			

This object contains individual batch information, like Batch ID created on Batch Close, number of transactions in a batch, etc.

9.2.1.2 BATCHDATA Message Format

XML Tag	Data Type / Max Length	Notes	Reference
<BATCHDATA>			
<BATCHCOUNT>	String	Number of batches	
<BATCHDETAILS>	Object Array	Array of BATCHDETAIL objects	BATCHDETAIL
<SECURENETID>	Integer / 8	7-digit unique SecureNet ID provided by SecureNet to identify the merchant and is required for each submitted transaction	
</BATCHDATA>			

Property Name	Type	Description	Reference
BATCHDATA	Object	Superset and Response object for Batches created	
BATCHDETAIL	Object	Subset of Batch data with Individual Batch Details Object is returned only if Batches are created and transactions exist to create a batch	

9.2.1.3 VAULTCUSTOMERRESPONSE

Please note that all fields in the VAULTCUSTOMERRESPONSE object are returned in a left-justified, space-filled format.

XML Tag	Data Type / Max Length	Notes	Reference
<VAULTCUSTOMERRESPONSE>			
<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 See Appendix	
<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>			

APPENDIX A: GATEWAY RESPONSE

A. Account Duplicate Check Indicator (ADCI)

Account Duplicate Check Indicator—Indicates whether duplicate account exists.

0—Does not check for Duplicate Card Number for specified Customer ID

1—Checks for Duplicate Card Number 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: *For ACCOUNT_VAULT only. Set to 0 by default.*

a. AVS_RESULT_CODE

Indicates the result of Address Verification System (AVS) checks.

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.*

b. 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
0	CVV/CID not provided
M	Match
N	No match
P	Not processed
S	Data not present
U	Issuer unable to process request
Y	Card Code Matches (Amex Only)

c. CARDLEVEL_RESULTS

This table provides additional information on card type:

NOTE: *MasterCard results are available only for SecureNet Frontend, First Data Nashville and TSYS processor.*

Code	Description
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 Travel Money
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

Code	Description
J1	General Prepaid
J2	Prepaid Gift Card
J3	Prepaid Healthcare

Provides additional information on MasterCard type:

Code	Description
DLG	Debit MasterCard Gold- Delayed Debit
DLH	Debit MasterCard World Embossed- Delayed Debit
DLI	Debit MasterCard Standard ISIC Student Card- Delayed Debit
DLS	Debit MasterCard Card- Delayed Debit
MAB	World Elite MasterCard Business
MAC	MasterCard Corporate World Elite
MBD	MasterCard Professional Debit Business Card
MBE	MasterCard Electronic Business Card
MBK	MasterCard Black
MCA	MasterCard Electronic Basic Card
MCB	MasterCard BusinessCard Card MasterCard Corporate Card
MCE	MasterCard Electronic Card
MCF	MasterCard Corporate Fleet Card
MCG	Gold MasterCard Card
MCH	MasterCard Premium Charge
MCM	MasterCard Corporate Meeting Card
MCO	MasterCard Corporate
MCP	MasterCard Corporate Purchasing Card
MCS	MasterCard Standard Card
MCT	Titanium MasterCard
MCU	MasterCard Unembossed
MCV	Merchant-Branded Program
MCW	World MasterCard Card
MDB	Debit MasterCard BusinessCard Card
MDG	Debit Gold MasterCard
MDH	World Debit Embossed
MDJ	Debit- Debit Other 2 Embossed
MDL	Business Debit Other Embossed
MDM	Middle Market Fleet Card
MDN	Middle Market Purchasing Card
MDO	Debit MasterCard Other
MDP	Debit MasterCard Platinum
MDQ	Middle Market Corporate Card
MDS	Debit MasterCard
MDT	MasterCard Business Debit

Code	Description
MDU	Debit MasterCard Unembossed
MEB	MasterCard Executive BusinessCard Card
MEC	MasterCard Electronic Commercial
MED	Debit MasterCard Electronic
MEF	MasterCard Electronic Payment Account
MEO	MasterCard Corporate Executive Card
MEP	Premium Debit MasterCard
MFB	Flex World Elite
MFD	Flex Platinum
MFE	Flex charge World Elite
MFH	Flex World
MFL	Flex Charge Platinum
MFW	Flex Charge World
MGF	MasterCard Government Commercial Card
MHA	MasterCard Healthcare Prepaid Non-tax
MHB	MasterCard HAS Substantiated
NHC	MasterCard Healthcare Credit Non-substantiated
MHH	MasterCard Unembossed Prepaid Student card
MIB	MasterCard Credit Electronic Student Card
MIC	MasterCard Credit Standard Student Card
MID	MasterCard Debit Unembossed Student Card
MIG	MasterCard Credit Umembossed Student Card
MIH	MasterCard Electronic Consumer Non U.S. Student Card
MIJ	MasterCard Debit Unembossed Non U.S. Student Card
MIK	MasterCard Electronic Consumer Prepaid Non U.S Student Card
MIL	MasterCard Unembossed Prepaid Non U.S Student Card
MIP	MasterCard Debit Prepaid Student Card
MIS	MasterCard Debit Standard Student Card
MIU	Debit MasterCard Unembossed Outside US
MLA	MasterCard Central Travel Solutions Air
MLC	MasterCard Micro-Business Card
MLD	MasterCard Distribution Card
MLL	MasterCard Central Travel Solutions Land
MNF	MasterCard Public Sector Commercial Card
MNW	World MasterCard Card (Europe)
MPA	Prepaid MasterCard Payroll Card
MPB	MasterCard Preferred BusinessCard
MPC	MasterCard Professional Card
MPF	Prepaid MasterCard Gift Card
MPG	Prepaid MasterCard Consumer Reloadable Card
MPJ	Prepaid Debit MasterCard Card Gold
MPK	Prepaid MasterCard Government Commercial Card

Code	Description
MPL	Platinum Mastercard Card
MPM	Prepaid MasterCard Consumer Promotion Card
MPN	Prepaid MasterCard Insurance Card
MPO	Prepaid MasterCard Other Card
MPR	Prepaid MasterCard Travel Card
MPT	Prepaid MasterCard Teen Card
MPV	Prepaid MasterCard Government Benefit Card
MPW	Prepaid MasterCard Corporate Card
MPX	Prepaid MasterCard Flex Benefit Card
MPY	Prepaid MasterCard Employee Incentive Card
MPZ	Prepaid MasterCard Emergency Assistance Card
MRB	Prepaid MasterCard Electronic BusinessCard
MRC	Prepaid MasterCard Electronic Card
MRF	Standard Deferred
MRG	Prepaid MasterCard Card Outside US
MRH	MasterCard Platinum Prepaid Travel Card
MRJ	Prepaid MasterCard Gold Card
MRL	Prepaid MasterCard Electronic Commercial
MRK	Prepaid MasterCard Public Sector Commercial Card
MRO	MasterCard Rewards Only
MRP	Standard Retailer Centric Payments
MRS	Prepaid MasterCard ISIC Student Card
MRW	Prepaid MasterCard BusinessCard Credit Outside US
MSD	Deferred Debit MasterCard
MUP	Premium Debit MasterCard Unembossed
MUS	Prepaid MasterCard Unembossed US
MUW	MasterCard World Domestic Affluent
MWB	World MasterCard for Business
MWD	World Deferred
MWE	MasterCard World Elite
MWO	MasterCard Corporate World
MWR	World Retailer Centric Payment
PVA	Private Label 1
PVB	Private Label 2
PVC	Private Label 3
PVD	Private Label 4
PVE	Private Label 5
PVF	Private Label 6
PVG	Private Label 7
PVH	Private Label 8
PVI	Private Label 9
PVJ	Private Label 10

Code	Description
SUR	Prepaid MasterCard Unembossed Outside US
TBE	Business – Immediate Debit
TCB	MasterCard Business Card—Immediate Debit
TCC	MasterCard (Mixed BIN) —Immediate Debit
TCE	MasterCard Electronic —Immediate Debit
TCF	MasterCard Fleet Card —Immediate Debit
TCG	Gold MasterCard Card —Immediate Debit
TCO	MasterCard Corporate —Immediate Debit
TCP	MasterCard Purchasing Card—Immediate Debit
TCS	MasterCard Standard Card —Immediate Debit
TCW	World Signia MasterCard Card —Immediate Debit
TDN	Middle Market MasterCard Purchasing Card —Immediate Debit
TEB	MasterCard Executive BusinessCard Card —Immediate Debit
TEC	MasterCard Electronic Commercial —Immediate Debit
TEO	MasterCard Corporate Executive Card —Immediate Debit
TIB	ISIC MasterCard Electronic Student Card —Immediate Debit
TIC	ISIC MasterCard Standard Student Card —Immediate Debit
TIU	MasterCard Unembossed —Immediate Debit
TLA	MasterCard Central Travel Solutions Air—Immediate Debit
TNF	MasterCard Public Sector Commercial Card —Immediate Debit
TNW	MasterCard New World —Immediate Debit
TPB	MasterCard Preferred Business Card —Immediate Debit
TPC	MasterCard Professional Card —Immediate Debit
TPL	Platinum MasterCard —Immediate Debit

d. CAVV_RESPONSE_CODE

Indicates the results of Cardholder Authentication Verification Value (CAVV) verification.

Code	Description
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

e. CARDTYPE Codes

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

CARDTYPE Code	Response Description	Reference
ACH	e-Check	
AX	American Express	
DB	Debit Card	
DC	Diners Club	
DS	Discover	
EBT	Electronic Benefits Transfer	
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	

f. DCI

Duplicate Transaction Indicator must be activated in Tools tab of the Virtual Terminal: "Duplicate Transaction Lockout Settings". Duplicates are evaluated against AMOUNT, CARDNUMBER, and, optionally, a user defined field (like Customer ID, Invoice Number or one of the UDFs), and within 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.*

B. 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

For Retail transactions—leave blank.

> ***NOTE:*** *If TRANSACTION_SERVICE = 1 or 2, INDUSTRYSPECIFICDATA will be set to 1 by default, unless transaction request specifies otherwise.*

C. METHOD

Method indicates the method of payment for the transaction being sent to the system and can be one of the following (The following values are case sensitive):

Code	Description	Reference
CC	Credit Card	
DB	Debit	
ECHECK	Electronic Check	
CHECK21	Electronic Check	
PD	PIN-less Debit	
SV	Stored Value	
EBT	Electronic Benefits Transfer	

a. 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 and only for the last 30 days. 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 0200-PRIOR_AUTH_CAPTURE. OrderID is limited to 25 characters; for example “CUSTOMERID MMddyyyyHHmmss”.

Format – Alphanumeric values only, No spaces and no special characters

b. REF_TRANSID

Transaction ID is a unique value provided by SecureNets Gateway in response for all transactions (approved and declined). This value can be used to reference a previously processed transaction for further processing like to void an approved transaction, capture a 0000-AUTH_ONLY transaction etc.

REF_TRANSID is required information for 0200-PRIOR_AUTH_CAPTURE, 0400-VOID Transaction.

RE-USE Account information to process a new transaction

It is possible to “re-use” account information (Credit Card, Bank account information, etc.) from an authorized transaction to [process a new transaction i.e. retrieve credit card information from a previously approved transaction and use this information to submit a new transaction. If the REF_TRANSID value is set in the TRANSACTION object using a previously approved transaction’s Transaction ID, a credit card number is not required to be sent, the credit card number from the transaction referenced by REF_TRANSID will be used. In order to re-use account information to process a new transaction, please follow the instructions below:

1. REF_TRANSID should be present. CARDNUMBER, ACCOUNTNUM, TRACKDATA, or MICRDATA should not be present.
2. Last 4 digit of card number or account number should be available
3. REF_TRANSID can be from a previously approved transaction.
4. REF_TRANSID should not be older than 90 days for the SecureNet ID.
5. Re-using account information from an approved transaction is available for the following transaction codes:
   a. 0100-AUTH_CAPTURE,0000-AUTH_ONLY,0300-CAPTURE_ONLY
   b. Re-using account information cannot be done for a Card-Present transaction, i.e. INDUSTRYSPECIFICDATA in the TRANSACTION object should not be empty or null, it should be set to 1, 2, 3, P, or D.

c. **RESPONSE_CODE**

Indicates the result of the transaction:

1—Approved
2—Declined
3—Error / Invalid Data

d. **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	Reference
ARC	Accounts Receivable Entries—A check received by a merchant through mail or drop box and presented as an ACH entry. Business Checking account, Personal Savings and Personal Checking accounts are only allowed	
BOC	Back Office Conversion—A check that is converted from paper to an electronic debit at a centralized location and presented as an ACH entry. Business Checking account, Personal Savings and Personal Checking accounts are only allowed	
CCD	Corporate Cash Disbursement—Primarily used for business to business transactions. Business Checking account is only allowed. Credits can only be processed using CCD for Business Accounts	
POS	Point-of-Sale—A debit at electronic terminal initiated by use of a plastic card. An example is using a Debit card to purchase gas. Personal Savings and Personal Checking accounts are only allowed	
PPD	Prearranged Payment and Deposits—Used to credit or debit a consumer account. Popularly used for payroll direct deposits and preauthorized bill payments. Personal Savings and Personal Checking accounts are only allowed. Credits can only be processed using PPD for Personal Accounts.	
TEL	Telephone Initiated-Entry—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). Personal Savings and Personal Checking accounts are only allowed	
WEB	Web Initiated-Entry—Electronic authorization through the Internet to create an ACH entry. Bank Account Holder First and Last Name are required. Personal Savings and Personal Checking accounts are only allowed	
POP	Business Checking account, Personal Savings and Personal Checking accounts are only allowed	

Please note that "Linked Credit transactions can be processed only if the original transaction used the following SEC codes; PPD, POS, WEB, TEL and CCD".

D. Transaction Type Table

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	Reference
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. Available for: ▪ E-checks* ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ StoredValueCard for Ceridian	
0001	PARTIAL_AUTH_ONLY	Used to validate a prepaid 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. Available for: ▪ Prepaid Cards ▪ Nashville platform only	
0100	AUTH_CAPTURE	Identical to AUTH_ONLY; however on approval the transaction will be picked up for settlement. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ PIN Debit ▪ PIN-less Debit ▪ Prepaid Cards ▪ StoredValueCard for Ceridian ▪ EBT	
0101	PARTIAL_AUTH_CAPTURE	Identical to PARTIAL_AUTH_ONLY; however on approval the transaction will be picked up for settlement. Available for: ▪ Prepaid Cards ▪ Credit Cards on TSYS and SecureNet FrontEnd platforms	

CODE	Transaction Type	Description	Reference
0200	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 to be settled. The transaction ID is valid and the system has a record of the original authorization-only transaction. The original transaction referred to is not already settled, expired, or voided. The amount being requested for settlement 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 originally authorized. Merchant account set up as Retail cannot capture an amount greater than the authorized amount (AUTH_ONLY to PRIOR_AUTH_CAPTURE). **NOTE:** 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. Available for: ▪ E-checks* ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ StoredValueCard for Ceridian	
0201	UPDATE_TRANS_INFO	To update Level II and Level III information for qualified transaction before settlement.	
0300	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. AUTHCODE is a required field for CAPTURE_ONLY transactions. Available for: ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ EBT	

CODE	Transaction Type	Description	Reference
0400	VOID	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 ▪ 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 original transaction to be voided (REF_TRANSID). The Gateway has a record of the transaction referenced by the ID. The transaction has not been sent for settlement. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ All platforms	
0401	PARTIAL_VOID	This action will reduce the amount of a previous unsettled transaction. The transaction is submitted with the ID of the original transaction to be voided (in REF_TRANSID field). The Gateway has a record of the transaction referenced by REF_TRANSID. The transaction has not been sent for settlement. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards MOTO and E-Commerce merchants only Nashville platform only, TSYS in pipeline	

CODE	Transaction Type	Description	Reference
0402	VOID_BY_OrderID	This action cancels a previous transaction such that it is not sent for settlement. It can be performed on unsettled transactions of the following types: ▪ AUTH_ONLY ▪ AUTH_CAPTURE ▪ PRIOR_AUTH_CAPTURE ▪ CREDIT_AUTHONLY ▪ FORCE_CREDIT. The transaction will be accepted by the Gateway if the following conditions are met: The transaction is submitted with the OrderID of the original transaction to be voided (in OrderID field). The Gateway has a record of the transaction referenced by the OrderID. The transaction has not been sent for settlement. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ All platforms	
0500	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 (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 (REF_TRANSID). The transaction ID is valid and the system has a record of the original transaction. The original transaction has been settled. 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. The card number (or the last four digits thereof) submitted with the credit transaction matches the card number of the original transaction. The credit transaction is submitted within 120 days of the settlement date and time of the original transaction. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ Debit cards for SecureNet Front End with Pin Data ▪ StoredValueCard for Ceridian ▪ EBT	
0501	CREDIT_AUTHONLY*	future user	
0502	CREDIT_PRIORAUTHCAPTURE*	future use	

CODE	Transaction Type	Description	Reference
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. Available for: ▪ E-checks ▪ Credit Cards ▪ Check Cards ▪ Prepaid Cards ▪ StoredValueCard for Ceridian ▪ EBT	
0601	FORCE_CREDIT_AUTHONLY*	future use	
0602	FORCE_CREDIT_PRIORAUTHCAPTURE*	future use	
0700	VERIFICATION	Run Transaction Verification on Credit Card or Bank Account with zero dollar amount. For Certegy ECHECK verification can be done for non-zero amount also. Available for: ▪ Nashville ▪ TSYS ▪ SecureNet FrontEnd platforms ▪ Credit Cards ▪ E-checks	
0800	AUTH_INCREMENT	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: ▪ The transaction is submitted with the ID of the transaction. ▪ The Gateway has a record of the transaction referenced by the ID. ▪ The transaction has not been sent for settlement. ▪ AUTH_INCREMENT can be submitted multiple times before submitting PRIOR_AUTH_CAPTURE.	
0900	ISSUE	This action issues a new card or adds funds to an already activated stored value card for paymentech platform and issues a card for Ceridian platform; also used for refunds to a stored value card for paymentech platform. Will be treated as credit to the card holder. Available for: ▪ Pre-paid Cards only ▪ Paymentech platform only ▪ Ceridian	
0901	ACTIVATE	This action activates a new card; only necessary if the stored value card is set to require activation. Available for: Pre-paid Cards Paymentech platform Ceridian	

CODE	Transaction Type	Description	Reference
0902	REDEEM	This action charges a card; the amount has to be within the available balance. Available for: ▪ Pre-paid Cards ▪ Paymentech platform ▪ Ceridian	
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. Available for: ▪ Pre-paid Cards only ▪ Paymentech platform only	
0904	DEACTIVATE	This action de-activates an activated card; amount should be specified as zero. Available for: ▪ Pre-paid Cards only ▪ Paymentech platform only	
0905	REACTIVATE	This action re-activates a de-activated card. Available for: ▪ Pre-paid Cards only ▪ Paymentech platform only	
0906	INQUIRY_BALANCE	This action checks the available balance on the stored value card; amount should be specified as zero. The available balance on the gift card will be returned in the TRANSACTIONAMOUNT field in the TRANSACTIONRESPONSE object. Available for: ▪ Pre-paid Cards only ▪ Paymentech platform only ▪ Ceridian	
0907	RELOAD/ RECHARGE	Adds value to a card Available for Ceridian*	
0908	ISSUE_VIRTUAL	This will issue a Virtual Card number and PIN. No plastic exists. Available for Ceridian*	
0909	CASH_OUT	This will withdraw all the available balance from the card. Available for Ceridian*	
0201	UPDATE TRANSACTION INFORMATION	This code enables updating Level2/Level3 transaction information of an approved transaction before it gets settled. Also this is used for CHECK21 image update and SIGNATURE image update. Both Image and Level2/Level3 information cannot be updated simultaneously.	

* Future use

E. SOFTDESCRIPTOR

Displays a description in addition to the merchants DBA. This is currently restricted to credit card transactions only. Contact Support to use this field. Maximum length is 25 characters.

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
Option 1:	Pos. 1–3:	Merchant name or abbreviation
	Pos. 4:	Asterisk (*)
	Pos. 5–25:	Descriptive Information
Option 2:	Pos. 1–7:	Merchant name or abbreviation
	Pos. 8:	Asterisk (*)
	Pos. 9–25:	Descriptive Information
Option 3:	Pos. 1–12:	Merchant name or abbreviation
	Pos. 13:	Asterisk (*)
	Pos. 14–25:	Descriptive Information

F. TRACKDATA

This field contains information encoded from a valid magnetic stripe and includes data such as Primary Account Number and Expiration Date. Unaltered Track Data must be forwarded intact. On Debit Card and EBT (Not Voucher Clear) transactions, this field is mandatory (with the exception of Debit Card PIN-Less transactions, which are entered manually). Track Data may contain Track 1 and/or Track 2 data. Track 2 data is required for Pin Debit and Stored Value transactions. Track 1 and Track2 data should not be submitted together. Gateway will also accept trackdata with beginning and end sentinel removed.

It is not acceptable to:

1. Use a manually entered account number and attempt to create a pseudo magnetic stripe.
2. Alter a Track 1 magnetic stripe and remove the cardholder name to make it look like a Track 2.
3. Flip the expiration date digits to MMYY order.
4. Truncate trailing spaces or zeroes.
5. Add, delete, or move characters in any way, including trailing spaces.

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

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

- The longitudinal redundancy checks (LRC) must be calculated for the data read from the Track and compared to the LRC read from the Track. The Track data is assumed to be read without errors when no character parity errors are detected and the calculated and read LRCs match.
- The starting sentinel, ending sentinel, and LRC are discarded.
- The character codes read from the magnetic stripe must be converted from the encoded character set to the set used for the authorization request message. The characters encoded on Track 1 are six bit plus parity codes and the characters encoded on Track 2 are four bit plus parity codes, with the character set

used for the request message defined as seven bit plus parity code. All characters read from a Track must be converted to the request message character set and transmitted as part of the request. The converted Track data cannot be modified by adding or deleting non-framing characters and must be a one for one representation of the characters read from the Track.

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

a. Track 1 Data

This is a variable length field with a maximum data length of 76 characters. The Track 1 data read from the cardholder's card is checked for parity and LRC errors and then converted from the six-bit characters encoded on the card to seven bit characters as defined in ANSI X3.4.

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

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

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

LEGEND:

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

b. Track 2 Data

This is a variable length field with a maximum data length of 37 characters. The Track 2 data read from the cardholder's card is checked for parity and LRC errors and then converted from the four-bit characters encoded on the card to seven bit characters as defined in ANSI X3.4. As part of the conversion, the terminal must remove the start sentinel, end sentinel, and LRC characters. The separators must be converted to either an ASCII "=" (HEX 3D) or ASCII "D" (HEX 44) characters. The entire UNALTERED Track (excluding framing characters) must be provided in the authorization request message or an error message will be generated.

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

SS	**PAN**	**FS**	**DATE**	**SVC CD**	**DISCRETIONARY DATA**	**ES**	**LRC**

LEGEND:

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

c. TRANSACTIONRESPONSE Additional Data Chart

Field Name	Debit Cards	EBT	Stored Value Cards	Fleet Cards	Reference
ADDITIONALDATA1			Previous Balance (15 characters long zero filled decimal number with explicit decimal point) followed by Cash Out Amount (15 characters long zero filled decimal number with explicit decimal point)		
ADDITIONALDATA2	Paymentech specific data—For Debit Transactions it will contain the Trace Number for the transaction. For PD transaction, settlement date is returned.	For EBT transactions it will contain the FNS # echoed from the ADDITIONALINFO field of the MERCHANT_KEY object.	for ISSUE_VIRTUAL: Virtual issued card number	Paymentech only—Optional data returned by the Host for Fleet Cards. (If data exists, it must be printed on the customer receipt as reference data).	
ADDITIONALDATA3		For EBT transactions it will contain the Voucher # echoed from the ADDITIONALINFO field of the CARD object.	for ISSUE_VIRTUAL: Virtual issued card EXP date		
ADDITIONALDATA4			for ISSUE_VIRTUAL: Virtual issued card Pin Number		

Field Name	Debit Cards	EBT	Stored Value Cards	Fleet Cards	Reference
ADDITIONALDATA5	P2P Encryption Transaction Response—String (22 Characters): 7-digit Key ID + {1,2} + MMddyyyyHHmmss	P2P Encryption Transaction Response—String (22 Characters): 7-digit Key ID + {1,2} + MMddyyyyHHmmss	P2P Encryption Transaction Response—String (22 Characters): 7-digit Key ID + {1,2} + MMddyyyyHHmmss	P2P Encryption Transaction Response—String (22 Characters): 7-digit Key ID + {1,2} + MMddyyyyHHmmss	
ADDITIONALAMOUNT	For Debit Transactions it will contain the surcharge for the transaction, if one has been setup on the Host. (If surcharge applies, it must be printed on the customer receipt). Applicable to Paymentech only.	Current Balance	Current Balance		
CASHBACK_AMOUNT	Cash Back Amount				
ADDITIONALAMOUNT	For Debit Transactions it will contain the surcharge for the transaction, if one has been setup on the Host. (If surcharge applies, it must be printed on the customer receipt). Applicable to Paymentech only.	Current Balance	Current Balance		

	CHECK Object	CARD Object		MERCHANT_KEY Object	Reference
ADDITIONAL INFO	ACH	Credit Card	EBT		
00	No Check Verification—(default value)	No credit card account verification, default value. Applicable only when adding a new Account to the SecureNet Vault.			Applicable to: ▪ Gateway Services ▪ Extended Services Supported Platform: ▪ Paymentech ▪ Ceridian ▪ ACH ▪ Refer: 7.00
01	Check verification through Certegy	Credit card account verification. Applicable only when adding a new Account to the SecureNet Vault.			
02	Pre-Note with ACH Provider	Update EXPDATE of credit card during Process Transaction by Customer ID.			

	CHECK Object	CARD Object		MERCHANT_KEY Object	Reference
Variable Text			11 (EBT indicator) + 15-character long left-justified space-filled EBT Voucher number. For EBT Voucher Clear transactions only.	For EBT transactions only. 11 (EBT indicator) + 7-digit FNS number (left-justified, space-filled, if FNS is less than 7 digits).	

d. TRANSACTION_SERVICE

[Regular Transactions]

0—Regular Transaction (Default)

[SecureNet Vault Transactions]

1—Process Transaction using Customer ID

2—Process Transaction using SECONDARY_MERCHANT_KEY

3—Process Transaction Add Customer and Account

G. Commodity Code for Level III items

Summary and Item Commodity code and Description fields should be populated with Standard United Nations Standard Products and Services Code(UNSPC) from unspsc.org

Standard codes can be found at http://www.unspsc.org/Search.asp

a. Unit of Measurement for Level III items

2 characters UOM should be submitted for Item Unit type for Level III transactions. Refer ANSI X-12 Allowable Units of Measure and Codes

Standard codes can be found at http://das.ct.gov/Purchase/Info/ANSI_UNITS_OF_MEASURE.pdf

b. HealthCare Code

2 characters HealthCare Code is required for HealthCare Transactions.

Refer United States Health Information Knowledgebase(USHIK) Metadata Registry for Accredited Standards Committee X12 Service Type Code.

Standard Service Type code can be found at http://ushik.ahrq.gov/ViewItemDetails?system=mdr&itemKey=27573000

c. Product Codes for Petroleum Industry Transactions

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:

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

APPENDIX B: DEVELOPER NOTES

Some best practices for SecureNet API implementation.

A. Gateway URL, SecureNet ID, SecureKey and test mode should be designed to be configurable by the developer who is integrating to the API

B. Using WSDL to process test and certification transactions

The below steps/considerations must be followed to add the correct WSDL to the project in order to test and certify the application:

1. The certify (test) WSDL is located in this URL: https://certify.securenet.com/api/gateway.svc
2. If Visual Studio is used, the WSDL reference will be added only as HTTP, not as HTTPS. This will cause a problem when trying to send transactions to the Gateway (live/production) environment using the same configuration.
3. Manually change all the references from HTTP to HTTPS in the code before running any test transactions
4. After adding reference in project, please use the end points below to run transactions:
   a. For SOAP: https://certify.securenet.com/API/Gateway.svc/soap (XML Web service)
   b. For XML Post: https://certify.securenet.com/API/Gateway.svc/web Http (XML post)

C. Using WSDL to process live/production transactions

The below steps/considerations must be followed to add the correct WSDL to the project in order to process live/production transactions once the application has been fully tested and certified by SecureNet:

1. Do not re-add any reference using a production URL - https://certify.securenet.com/api/gateway.svc
2. After adding reference in project, please use the end points below to run live/production transactions:
   a. For SOAP: https://gateway.securenet.com/API/Gateway.svc/soap (XML Web service)
   b. For XML Post: https://gateway.securenet.com/API/Gateway.svc/web Http (XML post)
3. While building a SOAP message, if wsdl is getting downloaded from certify and processing each transaction on production URL then it could cause issue if Demo is down or there is some mismatch between demo wsdl and production wsdl.

D. Order ID Implementation

Pre-generate the unique OrderID store it before it is submitted to the Gateway. In case of failure due to time out, non-receipt of response from the Gateway or unfinished transaction the original transaction can be retrieved by using the DCI or the transaction lookup method. Order ID will be matched only with approved transactions, the OrderID can be resubmitted for declined or error transactions. Order ID will be valid for a period of 30 days.

E. Response Handling

Business rules for Response Handling should be written based on Response Code and should not be based on Response Text. In case of Response Code 3, the Gateway does not have any record of the transaction.

F. Handling Transaction Time out

Applications should be designed to handle time out scenarios; in case of non-receipt of transaction response within the set application timeout period. Please consider the steps below:

- Set the transaction timeout for the application to 30 seconds
- Always send same Order ID if a time out occurs
- Use unique order ID across system
- Don’t re-attempt time out transaction within at least 60 seconds
- If a response code -3 and response reason code – 01O2 (for unique order id) is received, retrieve the original transaction response detail from the Gateway using extended API – “Get Transaction by OrderId”

DCI “Duplicate Transaction Indicator” can be very helpful to handle time outs, please consider below the steps below:

- Send DCI=2 on retried transaction if original transaction time’s out and predefine the settings in Virtual terminal, make sure to send all values as per setting.
- If retried transaction already exists as per the settings, the return response code will be “1” as approved and response reason code will be “DCT02”
- DCT02 is indicator of existing transaction from the Gateway only (customer is not charged twice)
- DCI indicator is not recommended if the business allows having same amount and same card for multiple transactions by same customer.
- Do not submit DCI for every transaction, submit DCI only in case of suspected duplicate or In case of failure due to time out, non-receipt of response from the Gateway

Time out for SecureNet Vault Accounts:

- CSDI and ACDI must be utilized to handle time out while performing SecureNet Vault functionality
- CSDI “Customer Duplicate Indicator” - CSDI=1 is recommended value; this will not add same customer again if customer is already exists.
- ACDI “Account Duplicate Indicator” – ACDI=1 is recommended value; this will not add same payment again if payment account already exists/associated with the customer record/customer ID.

G. Soft Decline

When merchant has turned on AVS and CVV based fraud settings, the transaction will be declined by the Gateway though it might be approved by the card issuer.

H. Global XSD

The Global XSD should be housed locally under the developer’s end point and should be updated only if required. Should be downloaded to a local file system of the application and configured to be utilized by the application. The XSD should not be validated against the online version to increase performance and reduce errors. During each change to the system, download the latest XSD to be utilized by the application.

I. Fault exceptions

- When submitting an incorrect request, which is not validated by the XSD, an “HTTP 400 Bad Request” response will be sent. Look at the RESPONSE_REASON_TEXT for description of the error and modify the request accordingly.
- When submitting an HTTP POST (REST) request:
  - Please make sure that all the request elements are present in the order that is defined in XSD.

  - Fields, which are of value type, i.e. Integer, Decimal, etc., must be present in the request even if not needed, if their parent object is present.

    For example: If TRANSACTION object is sent as request xml then OVERRIDE_FROM, TOTAL_INSTALLMENTCOUNT, INSTALLMENT_SEQUENCENUM, TRANSACTION_SERVICE, RETAIL_LANENUM, DCI, AMOUNT, CASHBACK_AMOUNT elements should be present in proper order (as defined in XSD) even if the values are 0.

- Transaction retry process timeouts, General Error Handling, response codes for retry
- Order ID
- Make configurable parameters for key API elements
- At the minimum Endpoint URL, SecureNet ID, SecureKey, Test Flag
- XSD implementation—copy locally and implement validation
- SecureNet Vault Layout
- SecureNet AutoBill Layout

J. Point to Point Encryption (P2PE)

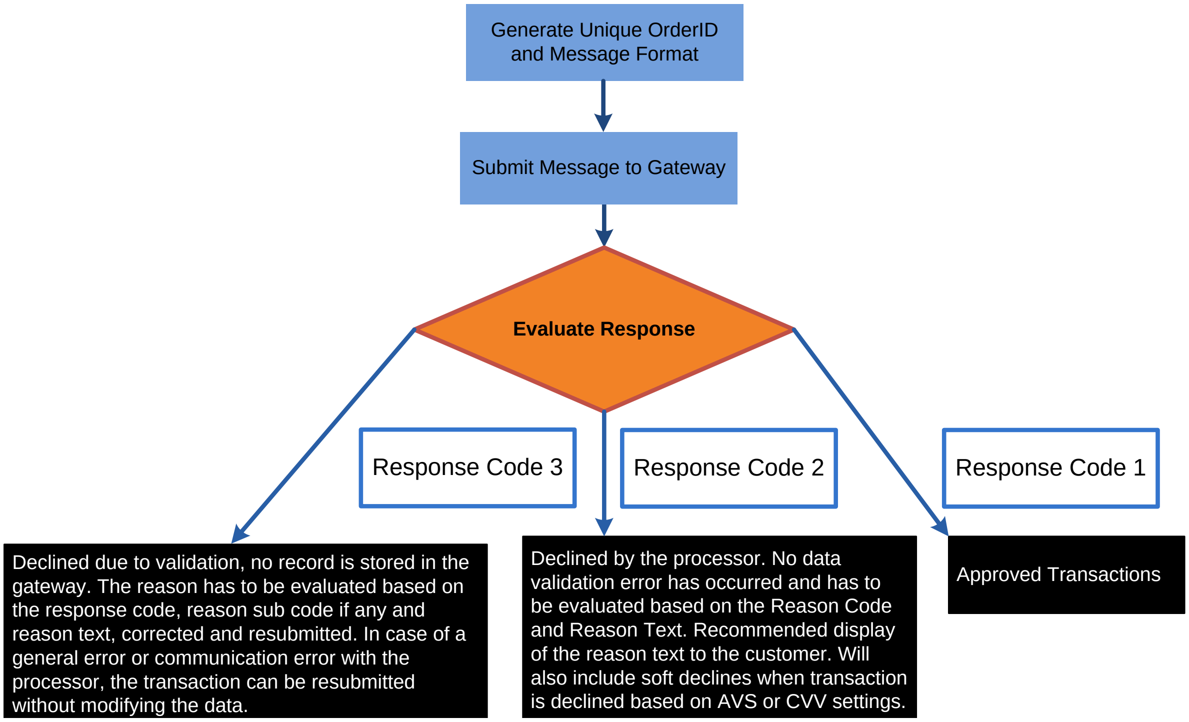
Hardware based encryption:

- SecureNet Gateway supports encrypted IDTech MSR’s
- The MSR’s has to be ordered with IDTech through SecureNet.
- The Injected Keys for the test and production are not the same

Software based encryption:

- Make sure to rotate keys on time of expiration
- Always keep primary and secondary key in active
- Do not create 2 keys within 2 days period and without proper validation (2 keys cannot be created in 24 hour period time)
- Make sure to process all transactions before new key is created.
- Check P2P key expiration date on transaction response field, create new key as soon as primary key is expired
- Prevent non encrypted data being resident in memory on the device executing the encryption software once the data is encrypted.
- Make sure PCI guidelines are followed.
- Refer “Gateway API P2PE Implementation Guide” for details on implementation

K. Response Handling Chart

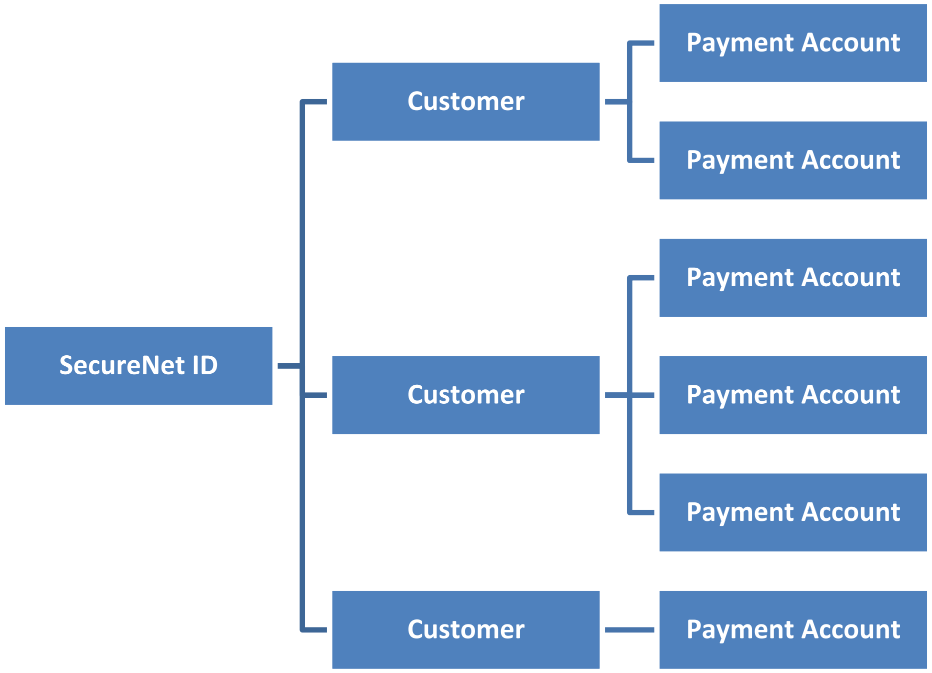


Declined due to validation, no record is stored in the gateway. The reason has to be evaluated based on the response code, reason sub code if any and reason text, corrected and resubmitted. In case of a general error or communication error with the processor, the transaction can be resubmitted without modifying the data.

Declined by the processor. No data validation error has occurred and has to be evaluated based on the Reason Code and Reason Text. Recommended display of the reason text to the customer. Will also include soft declines when transaction is declined based on AVS or CVV settings.

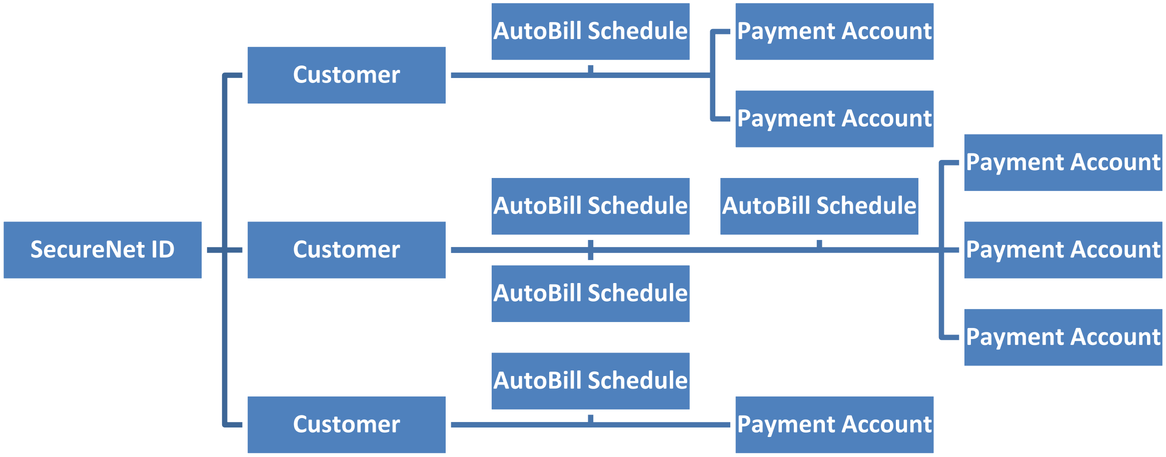
L. Tokenization

SecureNet Vault Customer and Payment Account Data Structure Examples



A token for an associated payment account is represent by combination customer ID and payment ID if the multiple forms of payment are stored per customer.

AutoBill / SecureNet Vault Customer / Payment Account and Schedule Data Structure Examples



M. 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 the sales agent, or send a request to support@securenet.com.

a. Credit Cards:

■ Test Credit Card Numbers for Approved Transactions

— American Express	370000000000002
— Discover	6011000000000012
— MasterCard	5424000000000015
— Visa	4444333322221111

■ 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

■ **Checks:**

— For checking account number—use any 6-digit number

— Valid Routing Numbers: 222371863
307075259
052000113

b. Code Samples

Contact SecureNet certification team/analyst for basic code sample for API implementation. Please note that the sample is for reference only to be utilized for application development.

i. API Error Codes

Gateway API can return multiple errors for different operations. Refer to the Error Codes Reference Guide document for a complete list (available upon request).

ii. How to Contact Technical Support

For problems with transaction processing or connection problems to the server, contact SecureNet Technical Support at support@securenet.com or (888)231-0060, option 3. Note that the SecureNet ID will be necessary and sample code might need to be provided to further research questions/problems.

NOTE: *API Support is available between the hours of 9am and 6pm Monday through Friday.*

iii. Certification Request Form

A certification request form is required, contact apiteam@securenet.com to request one if needed.

N. PCI Compliance

a. Storing and Protecting Data Beyond Authorization

Table 35 (along with information obtained from the PCI Data Security Standard, version 1.2) illustrates data elements commonly used for credit/debit card processing and the rules for storing and protecting these fields for use beyond the original authorization.

Table 35 Guidelines for storing and protecting sensitive data

	Data Element	Storage Permitted	Protection Required
Cardholder Data	Primary Account Number (PAN)*	Yes	Yes
	Cardholder Name**	Yes	Yes**
	Service Code**	Yes	Yes**
	Expiration Date**	Yes	Yes**
Sensitive Authentication Data****	Full Magnetic Stripe***	No	N/A
	CVD (CVC2/CVV2/CID)	No	N/A
	PIN/PIN Block	No	N/A

* The PAN (at a minimum) must be rendered unreadable anywhere it is stored.

** These data elements must be protected if stored in conjunction with the PAN.

*** Includes full track data from magnetic stripe, magnetic stripe image from the chip, or elsewhere.

**** Sensitive Authentication Data must not be stored after the authorization (even if encrypted).

For PCI-DSS and Payment Application Data Security Standard (PA-DSS) standards:
https://www.pcisecuritystandards.org/

O. General Processing Rules

This chapter covers general rules that apply to credit card processing.

a. Magnetic Stripe, Encrypted PIN, and CVD Retention

In accordance to the PCI Compliance Guidelines, it is strictly prohibited by to maintain the contents of a magnetic stripe or other cardholder verification information (such as the encrypted PIN or CVD) for future processing. Examples of this would be a recurring billing type transaction (such as monthly membership fees) or an additional authorization obtained at a hotel for an extended stay. You can store the account number and expiration date and send these types of recurring transactions for authorization as manually entered. To lower the potential for fraud, it is recommended to never store a magnetic stripe read after authorization is completed.

All certifications require that track and encrypted PIN data no longer be stored and submitted for specific card types following the initial authorization. For these card types, only the account number and expiration date should be delivered within any reversal advice requests or AFD completions.

See above, *PCI Compliance*, for detailed information relating to data retention guidelines.

P. Basic Card Verification Data

Visa and MasterCard implemented Card Verification Values (CVV) and Card Verification Codes (CVC2) to combat fraud. These values are the result of an algorithm and are encoded on the magnetic stripe of the card. Any alteration of the data elements read from the magnetic stripe prior to sending for authorization will cause the value not to match when the issuer reruns the algorithm at authorization time.

Refer to TRACKDATA details in Appendix A.

a. Enhanced Card Verification Data

The associations have implemented an enhanced security program that assists a merchant in the card-not-present environment to verify that the cardholder has possession of the card. Gateway refers to the data that is supported in this enhanced security program as CARDCODE.

Card verification programs

Card Type	Program Name	Length of Value	Card Placement
Visa	Enhanced Card Verification Value (CVV2)	3	Back of card, next to account number on signature panel
MasterCard	Enhanced Card Verification Code (CVC2)	3	Back of card, next to account number on signature panel
American Express	Card Identifier (CID)	4	Front of card, above on right of account number
Discover	Card Identifier (CID)	3	Back of card, next to account number on signature panel
Stored Value	Enhanced Card Verification Value (CVD2)	4	Back of card, typically hidden under scratch off panel

This information used is to determine whether or not a consumer is in possession of the physical card in a non-face-to-face environment. Although Gateway supports a CARDCODE on transactions that are both swiped and manually entered, most associations and issuers do not validate the CARDCODE when the card was swiped. When this happens, there will not be a response code in the transaction response.

Although Enhanced Card Verification Data is considered optional, for additional fraud protection, include this data in all manually keyed, card-not-present authorization requests.

The Gift Card program normally verifies the CARDCODE on manually entered redemptions, redemption auth-only, and balance inquiry transactions.

b. CARDCODE Response Information

The CARDCODE response can be used by the merchant to determine if the cardholder has possession of the card at the time of the sale. By using the CARDCODE response code to determine the legitimacy of the transaction, the merchant can reduce the potential of fraudulent transactions being processed, therefore reducing the merchants chargeback losses.

CARDCODE information does not impact interchange assessments.

See Appendix A, for a full listing of the CARDCODE response codes.

Q. Address Verification Service

Address Verification Service (AVS) provides merchants with additional information concerning the authentication of a particular transaction. Supporting AVS has several benefits for the merchant:

- Requesting to have the cardholder's address verified allows the merchant better fraud protection when the cardholder is not physically present.
- Requesting AVS may help to prevent chargebacks if the merchant can identify a fraudulent transaction prior to completing the sale/transaction.
- When processing transactions in the MOTO/Card Not Present/Direct Marketing industry or Electronic Commerce industry, Visa will assess a lower interchange rate if AVS is requested. (A match is not required.)
- When processing transactions in the Retail environment, merchants that must manually key an account number due to a bad magnetic stripe will receive a lower interchange rate if they request (and receive) verification of the cardholder's zip code. (A match is required.)
- AVS is provided by Visa, MasterCard, American Express, and Discover Cards. PO Box addresses are accepted if it is the cardholder's billing address.

See Appendix A for a full listing of AVS response code values by card type.

AVS-only requests are supported in Gateway to certain processors. When requesting address verification only, the transaction amount must equal $0.00 and has to be submitted as a verification transaction.

R. Commercial Card Processing

Both Visa and MasterCard offer businesses with a card platform that provides them with additional reporting so that the business can monitor their user activity on those cards. All of these cards fall under the umbrella term of *Commercial Cards*. The individual products are:

a. Corporate, Business, and Purchase

In order to provide this reporting to these businesses, merchants must submit additional data with each transaction. If submitted with the transaction, the merchant qualifies for lower consumer card interchange rates. If not submitted, both Visa and MasterCard assess higher Commercial interchange rates based on if the card was swiped or keyed, timeliness of the deposit, and so on.

1. Required Data Elements

Commercial Card data is required only at the time of settlement. Gateway Response includes indicator to specify if the card is Level II or Level III Valid. Merchant should be enabled to submit Level III data. If the transaction was submitted without any Level II or Level III data during the initial authorization the data can be updated before settlement using possible 2 methods.

If the transaction was submitted as AUTH_ONLY, the qualification data can be submitted during the PRIOR_AUTH_CAPTURE transaction.

If the transaction was submitted as AUTH_CAPTURE or PRIOR_AUTH_CAPTURE, UPDATE transaction can be submitted with the qualification data before settlement occurs.

Gateway supports both Level II and Level III data requirements as defined by Visa and MasterCard with certified processors only.

When processing Level III data, Level II data is required as well.

b. Account Truncation

Merchant must mask all of the cardholder’s account number (referred to as Account Truncation) except for the last four digits, and the expiration date must be masked in its entirety. This applies to:

- Both merchant and cardholder printed receipts
- Printed and displayed transaction reports
- Displayed transaction information

In addition, the merchant number or SecureNet ID must not be printed on the receipt.

Best practice would be to generate receipts in this manner whether online or printed.

Mask all of the cardholder’s account number (referred to as Account Truncation) except for the last four digits. This must be done by using either asterisk (*) or an X.

For example: ************1234 or XXXXXXXXXXXX1234

The same practice applies to ACH transactions also.

SecureNet recommendation that the cardholder's account number be masked and the expiration date is suppressed on ALL credit card receipts, regardless of the card type.

c. Recurring Payments

Visa, MasterCard, and Discover require that merchants indicate when a transaction is being processed on a recurring basis. All Recurring Transactions must be submitted to the host as Mail Order Transactions and the MOTO Indicator must equal **02**.

d. Electronic Commerce Processing Rules

Electronic Commerce environments are classified as those environments where the card and/or cardholder are not physically present at the time of purchase and the transactions are performed over the Internet. An e-commerce Web site accepting credit card payments for purchases made on their Web site would be an example.

Cardholders can only be billed for items actually shipped. If an item ordered turns out to be not in stock, the settled transaction should only be for the amount of the items shipped.

In order to obtain the best interchange rate for Electronic Commerce transactions, the following is required:

- Perform Auth_Only Transactions and capture when the item is shipped
- The transaction must be settled within 48 hours
- Industry-Specific Data must be present in all transactions
- The transaction amount must match the final authorized amount. **Amount tolerance between authorization and clearing is 0%.** (Partial void should be utilized to match the authorization and settlement amount
- The transaction date must match the ship date and must be no later than 7 days after the authorization date
- Address Verification (AVS) must be requested. A match is not required

e. Restaurant Processing Rules

In order to obtain the best interchange rate for Restaurant transactions, the following is required:

- The transaction must be settled within 24 hours
- Industry-Specific Data must be present in all transactions
- The settled transaction amount must be within 20% of the authorized amount. The initial authorization/sale must be submitted for the actual transaction amount. It must *not* be altered to include an additional percentage that accounts for a tip being added before settlement
- If the transaction must be manually keyed, AVS must be requested. A match of the zip code is required
- Cardholder signature is required

Tip Adjustment:

Transactions has to be submitted as AUTH_ONLY transaction, the tip should be included in the PRIOR_AUTH_CAPTURE before the batch close and should be added to the original sale amount in addition to the Tip Amount Field.

Tip re-adjustment can be done using the PRIOR_AUTH_CAPTURE before the batch is closed on the Gateway.

S. Batch Release Options

a. Auto Close

The Gateway will close the current open batch at a designated time each day.

This means that there is no end of day batch processing required from the merchant. Merchant is setup to auto close on the Gateway by default and has the ability to setup the close time based on their business needs.

***NOTE:** Verify with the merchant support and processor for the best cut off time to receive timely funding for the transactions*

Merchants with high volume of transactions should prefer to use the auto close option.

Multiple Batches might be created based upon the processor, For Example ACH would always be a separate batch or Stored Value will be a separate batch.

b. Manual Batch Close

The software provider can implement a function called a Batch Close using Transaction Services thus giving the merchant the flexibility to close the current open batch at any time throughout the day or perform multiple Batch Releases.

Also manual batch close can be performed on the Virtual Terminal using unsettled transactions report.

c. Batch Inquiry

Batch Inquiry functionality using Extended Services allows the POS to receive transaction details (all transactions or by card type) or card type totals (all card types or a specific card type) for a particular Batch ID.

***NOTE:** Manual Batch Release and Batch Inquiry is not recommended for merchant having more than 1000 transactions per day.*

APPENDIX C: REFERENCES

A. Introduction

The Valutec WS is a standard web service as defined by World Wide Web Consortium (W3C, http://www.w3.org).

It provides several methods that the programs and web applications can use to process transactions, provide reports, and update customer information on Valutec gift and loyalty cards.

If not familiar with web services, examine the following resources:

- For an overview of web services and related technology: http://en.wikipedia.org/wiki/Web_service
- For an introduction to XML, the foundation for both SOAP & WSDL: http://www.w3schools.com/xml/default.asp
- For a tutorial on SOAP, the communication language of web services: http://www.w3schools.com/soap/default.asp
- For a tutorial on WSDL, the language that web services use: http://www.w3schools.com/wsdl/default.asp

a. Automatic Discovery and Code Generation

The code documents itself in a form that is machine readable, using WSDL. You can obtain the WSDL definition of this web service with the links provided for the appropriate service.

Many Java and .NET development environments include or have available tools that can read this WSDL and automatically generate a set of classes that encapsulate all the details of interacting with the web service. You can then use the web service via these wrapper classes without needing to deal with any of the WSDL, SOAP, HTTP, or network communications details. After the wrapper classes have been generated, the only code that needs to be written is to instantiate the wrapper and call the approprithete transaction method on it. Only 1 or 2 lines of code will be required to be written (thanks to the hundreds of lines that were already automatically generated).

If these web services tools are available, there only two things that need to be known in order to start using this web service:

1. How to use web service tools
2. The WSDL URL for this web service

NOTE: *The names of some web methods may be modified by existing tools in order to follow naming conventions or avoid the illegal use of keywords in the programming language. For example, in Java and other languages the void(...) web method may be renamed to _void(...) or some other variation because void is a language keyword.*

If these web services tools are not available, then use one of the other options:

- If comfortable with SOAP, use the SOAP 1.1 or SOAP 1.2 interfaces. The documentation for each web method (links provided below) contains sample SOAP 1.1 and 1.2 request/response pairs. Note that each sample shows the entire HTTP/S message including header fields and content.
  - — Production SOAP URLs
    - https://gateway.securenet.com/API/Gateway.svc/soap, for Transaction API (Gateway services)
    - https://gateway.securenet.com/API/data/service.svc/soap, Extended Transaction Functionality API (Extended services)

    - https://gateway.securenet.com/API/data/transaction.svc/soap, for Extended Transaction Process API (Transaction services)
  - Certification SOAP URLs
    - https://certify.securenet.com/API/Gateway.svc/soap, for Transaction API (Gateway services)
    - https://certify.securenet.com/API/data/service.svc/soap, for Extended Transaction Functionality API (Extended services)
    - https://certify.securenet.com/API/data/transaction.svc/soap, for Extended Transaction Process API (Transaction services)
- Use the HTTP GET interfaces. Sample request/response pairs for this interface are provided with the documentation for each web method. Note that each sample shows the entire HTTP/S message including header fields and content.
- If not using tool-generated wrappers, be familiar and comfortable with the libraries for raw TCP/IP and/or HTTPS communication.

Revision History

Date	Version	Revision	Description of change(s)	Author(s)
12/23/2011	4.1.2	1	Initial Document	Preetham Gowda, Jorge Tardio, Narendra Solanki
01/11/2012	4.1.2	2	Added UpdateTransaction Method to Section 4.1	Narendra Solanki
01/11/2012	4.1.2	2	Added Section 4.7 – Method for Batch Close	Narendra Solanki
01/11/2012	4.1.2	2	Added Section 5.4 – Transaction Image Request	Narendra Solanki
01/11/2012	4.1.2	2	Added Section 9.1.1.6 - IMAGERESPONSE	Narendra Solanki
01/27/2012	4.1.2	2	Modified verbiage for REF_TRANSID in Appendix	Jorge Tardio, Narendra Solanki
5/03/2012	4.1.3	3	Added to Sections: 8.20.2, 8.20.3 and 8.21.1.1 Added to Appendix A, Section C Added to Appendix B, Section C	Mukesh Agrawal
6/21/2012	4.1.3	3	Revised Section 4.5--Reporting Methods for SecureNet Vault and AutoBill Services [Extended Services], GetABAccounts—Method to retrieve all AutoBill accounts of CustomerID	Narendra Solanki
08/20/2012	4.1.4	4	Updated CARDLEVEL_RESULTS values in Appendix A for MasterCard	Kathiravan Srinivasan
08/20/2012	4.1.4	4	Revised CARDLEVEL_RESULTS values in Appendix A for MasterCard	Narendra Solanki
8/22/2012	4.1.4	4	Formatted/edited new additions. Added Proprietary and Confidential to cover and footer, updated TOC and version.	Cindy Magee
9/14/2012	4.1.4	4	Revised and updated Hotel Sale Code, Charge Type and Auto Sale Code. Fixed bookmarked links.	Stella Wateba/Cindy Magee