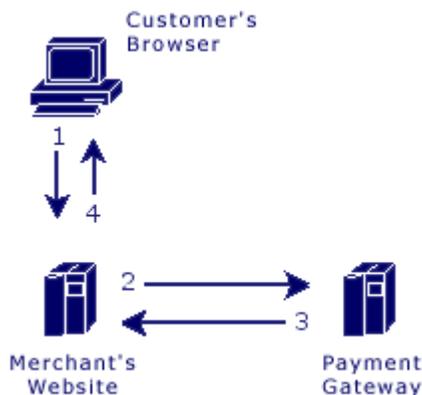


# Methodology

## Direct Post API

### Transactions



#### Steps:

1. The customer sends their payment information to the merchant's web site.
2. The merchant web site posts the payment data to the Payment Gateway.
3. The Payment Gateway responds immediately with the results of the transactions.
4. The merchant web site displays the appropriate message to the customer.

The communication method used to send messages to the Payment Gateway's server is the standard HTTP protocol over an SSL connection.

In the Direct Post method, the communications with the cardholder (Steps 1 and 4) are developed completely by the merchant and therefore are not defined by the Payment Gateway. Step 1 should simply collect the payment data from the cardholder and Step 4 should display the appropriate transaction receipt or declined message.

In Step 2, transaction details should be delivered to the Payment Gateway using the POST method with the appropriate variables defined below posted along with the request.

In Step 3, the transaction responses are returned in the body of the HTTP response in a query string name/value format delimited by ampersands. For example: `variable1=value1&variable2=value2&variable3=value3`

### Customer Vault

The Customer Vault was designed specifically for businesses of any size to address concerns about handling customer payment information. Visa and MasterCard have instituted the Payment Card Industry (PCI) Data Security to protect cardholder data, wherever it resides, ensuring that members, merchants, and service providers maintain the highest information security standards.

These associations have also deemed that merchants will be held liable for any breach of cardholder data. This has become a major concern for merchants who handle credit card or electronic check payments. The Customer Vault is designed for these merchants who desire to avoid the tremendous costs and resources involved in becoming PCI compliant under these circumstances.

The Customer Vault does this by allowing merchants to transmit their payment information through a Secure Sockets Layer (SSL) connection for storage in our Level 1 PCI certified data facility. Once the customer record has been securely transmitted to the Customer Vault, the merchant can then initiate transactions remotely without having to access cardholder information directly. This process is accomplished without the merchant storing the customer's payment information in their local database or payment application.

## **Transaction Types**

### **Direct Post API**

#### **Sale (sale)**

Transaction sales are submitted and immediately flagged for settlement.

#### **Authorization (auth)**

Transaction authorizations are authorized immediately but are not flagged for settlement. These transactions must be flagged for settlement using the capture transaction type. Authorizations typically remain active for three to seven business days.

#### **Capture (capture)**

Transaction captures flag existing authorizations for settlement. Only authorizations can be captured. Captures can be submitted for an amount equal to or less than the original authorization.

#### **Void (void)**

Transaction voids will cancel an existing sale or captured authorization. In addition, non-captured authorizations can be voided to prevent any future capture. Voids can only occur if the transaction has not been settled.

#### **Refund (refund)**

Transaction refunds will reverse a previously settled transaction. If the transaction has not been settled, it must be voided instead of refunded.

## Credit (credit)

Transaction credits apply an amount to the cardholder's card that was not originally processed through the Gateway. In most situations credits are disabled as transaction refunds should be used instead.

## Validate (validate)

This action is used for doing an "Account Verification" on the cardholder's credit card without actually doing an authorization.

## Update (update)

Transaction updates can be used to update previous transactions with specific order information, such as a tracking number and shipping carrier.

# Transaction Variables

## Direct Post API

### Sale/Authorization/Credit/Validate

Variable Name	Description
type*	The type of transaction to be processed. Values: 'sale', 'auth', 'credit', or 'validate'
username*	Username assigned to merchant account.
password*	Password for the specified username.
ccnumber**	Credit card number.
ccexp**	Credit card expiration date. Format: MMY
cvv	The card security code. While this is not required, it is strongly recommended.
checkname***	The name on the customer's ACH account.
checkaba***	The customer's bank routing number.
checkaccount***	The customer's bank account number.
account_holder_type***	The type of ACH account the customer has. Values: 'business' or 'personal'
account_type***	The ACH account entity of the customer. Values: 'checking' or 'savings'

sec_code***	The Standard Entry Class code of the ACH transaction. Values: 'PPD', 'WEB', 'TEL', or 'CCD'
amount	Total amount to be charged. For validate, the amount must be omitted or set to 0.00. Format: x.xx
currency	The transaction currency. Format: ISO 4217
payment***	The type of payment. Default: 'creditcard' Values: 'creditcard' or 'check'
processor_id	If using Multiple MIDs, route to this processor (processor_id is obtained under Settings->Load Balancing in the Control Panel).
dup_seconds	Sets the time in seconds for duplicate transaction checking on supported processors. Set to 0 to disable duplicate checking.
descriptor	Set payment descriptor on supported processors.
descriptor_phone	Set payment descriptor phone on supported processors.
descriptor_address	Set payment descriptor address on supported processors.
descriptor_city	Set payment descriptor city on supported processors.
descriptor_state	Set payment descriptor state on supported processors.
descriptor_postal	Set payment descriptor postal code on supported processors.
descriptor_country	Set payment descriptor country on supported processors.
descriptor_mcc	Set payment descriptor mcc on supported processors.
descriptor_merchant_id	Set payment descriptor merchant id on supported processors.
descriptor_url	Set payment descriptor url on supported processors.
billing_method	Should be set to 'recurring' to mark payment as a recurring transaction. Values: 'recurring'
orderdescription	Order description.
orderid	Order Id
ipaddress	IP address of cardholder, this field is recommended. Format: xxx.xxx.xxx.xxx
tax****	Total tax amount.
shipping****	Total shipping amount
ponumber****	Original purchase order
firstname	Cardholder's first name.
lastname	Cardholder's last name
company	Cardholder's company
address1	Card billing address
address2	Card billing address, line 2
city	Card billing city
state	Card billing state. Format: CC
zip	Card billing zip code
country	Card billing country. Country codes are as shown in ISO 3166. Format: CC

phone	Billing phone number
fax	Billing fax number
email	Billing email address
social_security_number	Customer's social security number, checked against bad check writers database if check verification is enabled.
drivers_license_number	Driver's license number.
drivers_license_dob	Driver's license date of birth.
drivers_license_state	The state that issued the customer's driver's license.
validation	Specify which Validation processors to use. If variable is passed blank, use none.
shipping_firstname	Shipping first name
shipping_lastname	Shipping last name
shipping_company	Shipping company
shipping_address1	Shipping address
shipping_address2	Shipping address, line 2
shipping_city	Shipping city
shipping_state	Shipping state Format: CC
shipping_zip	Shipping zip code
shipping_country	Shipping country Country codes are as shown in ISO 3166. Format: CC
shipping_email	Shipping email address
merchant_defined_field_#	You can pass custom information in up to 20 fields. Format: merchant_defined_field_1=Value
recurring	Recurring action to be processed. Values: add_subscription
plan_id	Create a subscription tied to a Plan ID if the sale/auth transaction is successful.
plan_payments	The number of payments before the recurring plan is complete. Note: Use '0' for 'until canceled'
plan_amount	The plan amount to be charged each billing cycle. Format: x.xx
day_frequency	How often, in days, to charge the customer. Cannot be set with 'month_frequency' or 'day_of_month'.
month_frequency	How often, in months, to charge the customer. Cannot be set with 'day_frequency'. Must be set with 'day_of_month'. Values: 1 through 24
day_of_month	The day that the customer will be charged. Cannot be set with 'day_frequency'. Must be set with 'month_frequency'. Values: 1 through 31 - for months without 29, 30, or 31 days, the charge will be on the last day
start_date	The first day that the customer will be charged. Format: YYYYMMDD

### **Level III specific order fields**

shipping	Freight or shipping amount included in the transaction amount Default: '0.00' Format: x.xx
tax	The sales tax, included in the transaction amount, associated with the purchase. Setting tax equal to '-1' indicates an order that is exempt from sales tax. Default: '0.00' Format: x.xx
ponumber†	Purchase order number supplied by cardholder
orderid†	Identifier assigned by the merchant. This defaults to gateway transaction id.
shipping_country†	Shipping country (e.g. US) Format: CC
shipping_postal†	Postal/ZIP code of the address where purchased goods will be delivered. This field can be identical to the 'ship_from_postal' if the customer is present and takes immediate possession of the goods.
ship_from_postal†	Postal/ZIP code of the address from where purchased goods are being shipped, defaults to merchant profile postal code.
summary_commodity_code†	4 character international description code of the overall goods or services being supplied. The acquirer or processor will provide a list of current codes.
duty_amount	Amount included in the transaction amount associated with the import of purchased goods. Default: '0.00' Format: x.xx
discount_amount	Amount included in the transaction amount of any discount applied to complete order by the merchant. Default: '0.00' Format: x.xx
national_tax_amount	The national tax amount included in the transaction amount. Default: '0.00' Format: x.xx
alternate_tax_amount	Second tax amount included in the transaction amount in countries where more than one type of tax can be applied to the purchases. Default: '0.00' Format: x.xx
alternate_tax_id	Tax identification number of the merchant that reported the alternate tax amount.
vat_tax_amount	Contains the amount of any value added taxes which can be associated with the purchased item. Default: '0.00' Format: x.xx
vat_tax_rate	Contains the tax rate used to calculate the sales tax amount appearing. Can contain up to 2 decimal places, e.g. 1% = 1.00. Default: '0.00' Format: x.xx
vat_invoice_reference_number	Invoice number that is associated with the VAT invoice.

customer_vat_registration	Value added tax registration number supplied by the cardholder.
merchant_vat_registration	Government assigned tax identification number of the merchant for whom the goods or services were purchased from.
order_date	Purchase order date, defaults to the date of the transaction. Format: YYMMDD

### Level III specific line item detail fields

item_product_code_#†	Merchant defined description code of the item being purchased.
item_description_#†	Description of the item(s) being supplied.
item_commodity_code_#†	International description code of the individual good or service being supplied. The acquirer or processor will provide a list of current codes.
item_unit_of_measure_#†	Code for units of measurement as used in international trade. Default: 'EACH'
item_unit_cost_#†	Unit cost of item purchased, may contain up to 4 decimal places.
item_quantity_#†	Quantity of the item(s) being purchased. Default: '1'
item_total_amount_#†	Purchase amount associated with the item. Defaults to: 'item_unit_cost_#' x 'item_quantity_#' rounded to the nearest penny.
item_tax_amount_#†	Amount of tax on specific item, amount should not be included in 'total_amount_#'. Default: '0.00'
item_tax_rate_#†	Percentage representing the value-added tax applied. Default: '0.00'
item_discount_amount_#	Discount amount which can have been applied by the merchant on the sale of the specific item. Amount should not be included in 'total_amount_#'.
item_discount_rate_#	Discount rate for the line item. 1% = 1.00. Default: '0.00'
item_tax_type_#	Type of value-added taxes that are being used.
item_alternate_tax_id_#	Tax identification number of the merchant that reported the alternate tax amount.

- \* Always required
- \*\* Required for credit card transactions
- \*\*\* Required for ACH transactions
- \*\*\*\* Required for Level 2 transactions
- † Required for Level 3 transactions

#### Notes:

- Level II fields are required for Level II processing.
- Level II and Level III fields are required for Level III processing.
- You can pass only credit card **or** e-check transaction variables in a request, not both in the same request.
- Certain banks may require some optional fields.

## Capture

**Variable Name**

**Description**

type*	Type of transaction. Values: 'capture'
username*	Username assigned to merchant account.
password*	Password for the specified username.
transactionid*	Original payment gateway transaction id
amount*	Total amount to be settled. This amount must be equal to or less than the original authorized amount. Format: x.xx
tracking_number	Shipping tracking number
shipping_carrier	Shipping carrier. Values: 'ups', 'fedex', 'dhl', or 'usps'
orderid	Order id.

\* Always required

## Void

Variable Name	Description
type*	Type of transaction. Values: 'void'
username*	Username assigned to merchant account.
password*	Password for the specified username.
transactionid*	Original payment gateway transaction id

\* Always required

## Refund

Variable Name	Description
type*	Type of transaction. Values: 'refund'
username*	Username assigned to merchant account.
password*	Password for the specified username.
transactionid*	Original payment gateway transaction id
amount	Total amount to be refunded. This amount may be equal to or less than the settled amount. Format: x.xx

\* Always required

## Update

Variable Name	Description
type*	Type of transactions. Values: 'update'
username*	Username assigned to merchant account.
password*	Password for the specified username.
transactionid*	Original payment gateway transaction id

tracking_number	Shipping tracking number
shipping	Total shipping amount. Format: x.xx
shipping_postal	Postal/ZIP code of the address where purchased goods will be delivered. This field can be identical to the 'ship_from_postal' if the customer is present and takes immediate possession of the goods.
ship_from_postal	Postal/ZIP code of the address from where purchased goods are being shipped, defaults to merchant profile postal code.
shipping_country	Shipping Country Code.
shipping_carrier	Shipping carrier. Values: 'ups', 'fedex', 'dhl', or 'usps'
shipping_date	Shipping date. Format: YYYYMMDD
orderdescription	Order Description.
order_date	Order date. Format: YYYYMMDD
customer_receipt	If set to true, when the customer is charged, they will be sent a transaction receipt. Values: 'true' or 'false'
ponumber	Cardholder's purchase order number.
summary_commodity_code	4 character international description code of the overall goods or services being supplied. The acquirer or processor will provide a list of current codes.
duty_amount	Amount included in the transaction amount associated with the import of purchased goods. Format: x.xx
discount_amount	Amount included in the transaction amount of any discount applied to complete order by the merchant. Format: x.xx
tax	Tax amount. Format: x.xx
national_tax_amount	The national tax amount included in the transaction amount. Format: x.xx
alternate_tax_amount	Second tax amount included in the transaction amount in countries where more than one type of tax can be applied to the purchases. Format: x.xx
alternate_tax_id	Tax identification number of the merchant that reported the alternate tax amount.
vat_tax_amount	Contains the amount of any value added taxes which can be associated with the purchased item.
vat_tax_rate	Contains the tax rate used to calculate the sales tax amount appearing. Can contain up to 2 decimal places, e.g. 1% = 1.00.
vat_invoice_reference_number	Invoice number that is associated with the VAT invoice.
customer_vat_registration	Value added tax registration number supplied by the cardholder.

merchant_vat_registration	Government assigned tax identification number of the merchant for whom the goods or services were purchased from.
merchant_defined_field_#	Merchant Defined Fields. Format: merchant_defined_field_1=Value

\* Always required

## Retail Data

### Direct Post API

#### Passing Unencrypted Retail Magnetic Stripe Data

Variable Name	Description
track_1	Raw Magnetic Stripe Data
track_2	Raw Magnetic Stripe Data
track_3	Raw Magnetic Stripe Data

#### Passing MagTek Magensa Encrypted Magnetic Stripe Data

Variable Name	Description
magnesafe_track_1	Raw MagTek Magensa Data
magnesafe_track_2	Raw MagTek Magensa Data
magnesafe_magneprint	Raw MagTek Magensa Data
magnesafe_ksn	Raw MagTek Magensa Data
magnesafe_magneprint_status	Raw MagTek Magensa Data

#### Passing IDTech M130 Encrypted Swipe Data

Variable Name	Description
encrypted_track_1	Raw encrypted data
encrypted_track_2	Raw encrypted data
encrypted_track_3	Raw encrypted data
encrypted_ksn	Raw encrypted data

#### Passing IDTech M130 Encrypted Keyed Data

Variable Name	Description
encrypted_data	Raw encrypted data

## Apple Pay

### Direct Post API

# Supported Processors

Currently Apple Pay is supported only on the Vital platform.

## Configuring Apple Pay

### Creating an Apple Merchant ID

First, you must obtain an Apple Merchant ID before you can generate the Certificate Signing Request that Apple requires. You will need to set up an Apple Merchant ID in your iOS Developer Account. Follow these steps to complete the setup:

1. Go to Apple's Developer Portal and log in to the Member Center to create a new Merchant ID.
2. Navigate to the Certificates, Identifiers, and Profiles area of the Member Center, and then begin the Register Merchant ID process.
3. You must then set the Apple Merchant ID within your gateway Control Panel under Settings -> Apple Pay.

### Generating the Certificate Signing Request

Next, you will need to associate a Certificate with the Merchant ID in Apple's Developer Portal. After downloading the Certificate Signing Request from the gateway's options page, follow these steps.

1. In Apple's Developer Portal, click on the Merchant ID and then click "Edit".
2. Click "Create Certificate".
3. You are obtaining a CSR file from a Payment Provider so you will not have to create one. Click "Continue" to proceed to the upload page.
4. Click "Choose File..." and select the Gateway.certSigningRequest file you downloaded from the gateway's options page.

## How to Obtain Apple Pay Payment Data

[PassKit](#) provides the payment data in the (*PKPayment \**)*payment* that is returned to your app's *paymentAuthorizationViewController:didAuthorizePayment:completion* method. The Apple Pay encrypted payment data is found in *payment.token.paymentData*.

*payment.token.paymentData* is a binary (NSData) object, so you must encode it as a hexadecimal string before it can be passed to the Gateway.

## Passing Apple Pay Payment Data

To submit a payment with Apple Pay, send the encrypted token data into the *applepay\_payment\_data* variable. There is no need to decrypt the data in your app. Only the Gateway will have access to the private key that can decrypt the token.

## Notes

When passing in *applepay\_payment\_data*, you should not include the variables *ccnumber* or *ccexp*; they are

extracted from the token data.

**Important Note:** The authorization amount must match the amount the customer approves in the app. If you pass in a currency, that must also match the currency approved in the app. If omitted, the currency from the app is used.

For working example code, including how to obtain the PKPayment object and how to pass a simple transaction to the Gateway, [download the sample project](#).

## Variables

Variable Name	Description
applepay_payment_data	The encrypted Apple Pay payment data (payment.token.paymentData) from PassKit encoded as a hexadecimal string

## Troubleshooting

If you receive the error "Failed to decrypt Apple Pay data. Ensure that the Apple Pay Merchant ID is correct in the Gateway Settings and that the certificate was generated from a Gateway Certificate Signing Request.", try these steps:

1. Verify that the Merchant ID that Apple has in the developer portal exactly matches the Merchant ID in the Gateway's settings.
2. Verify that your app's PKPaymentRequest's merchantIdentifier exactly matches the Merchant ID in the Gateway's settings.
3. Ensure that the correct Merchant ID is checked in the Apple Pay section of the Capabilities tab in your project's target settings.
4. Try creating a new Merchant ID. Reusing an existing Merchant ID with a new certificate may sometimes cause issues with encryption.

## Recurring Variables

### Direct Post API

#### Add a Plan

Variable Name	Description
recurring*	Add a recurring plan that subscriptions can be added to in the future. Value: 'add_plan'
plan_payments*	The number of payments before the recurring plan is complete. Notes: '0' for until canceled

plan_amount*	The plan amount to be charged each billing cycle. Format: x.xx
plan_name*	The display name of the plan.
plan_id*	The unique plan ID that references only this recurring plan.
day_frequency**	How often, in days, to charge the customer. Cannot be set with 'month_frequency' or 'day_of_month'.
month_frequency***	How often, in months, to charge the customer. Cannot be set with 'day_frequency'. Must be set with 'day_of_month'. Values: 1 through 24
day_of_month***	The day that the customer will be charged. Cannot be set with 'day_frequency'. Must be set with 'month_frequency'. Values: 1 through 31 - for months without 29, 30, or 31 days, the charge will be on the last day
* Always required	
** Required unless 'month_frequency' and 'day_of_month' is set.	
*** Required unless 'day_frequency' is set.	

## Add a Subscription to an Existing Plan

Variable Name	Description
recurring*	Associate payment information with a recurring plan. Value: add_subscription
plan_id*	The plan ID of the plan that the subscription will be associated with.
start_date	The first day that the customer will be charged. Format: YYYYMMDD
ccnumber**	Credit card number.
ccexp**	Credit card expiration. Format: MMY
payment***	The type of payment. Default: 'creditcard' Values: 'creditcard' or 'check'
checkname***	The name on the customer's ACH account.
checkaccount***	The customer's bank account number.
checkaba***	The customer's bank routing number.
account_type***	The customer's ACH account type. Values: 'checking' or 'savings'
account_holder_type***	The customer's ACH account entity. Values: 'personal' or 'business'
sec_code	ACH standard entry class codes. Values: 'PPD', 'WEB', 'TEL', or 'CCD'
first_name	Cardholder's first name.
last_name	Cardholder's last name.
address1	Card billing address.
city	Card billing city

state	Card billing state.
zip	Card billing postal code.
country	Card billing country code.
phone	Billing phone number.
email	Billing email address.
company	Cardholder's company.
address2	Card billing address, line 2.
fax	Billing fax number.
orderid	Order ID
order_description	Order Description
merchant_defined_field_#	Can be set up in merchant control panel under 'Settings'->'Merchant Defined Fields'.
ponumber	Cardholder's purchase order number.
processor_id	If using Multiple MIDs, route to this processor (processor_id is obtained under Settings-&gt;Load Balancing in the Control Panel).
customer_receipt	If set to true, when the customer is charged, they will be sent a transaction receipt. Values: 'true' or 'false'
* Always required	
** Required for credit card transactions	
*** Required for ACH transactions	

## Adding a Custom Subscription

Variable Name	Description
recurring*	Add a custom recurring subscription that is NOT associated with an existing plan Value: 'add_subscription'
plan_payments*	The number of payments before the recurring plan is complete. Notes: '0' for until canceled
plan_amount*	The plan amount to be charged each billing cycle. Format: x.xx
day_frequency**	How often, in days, to charge the customer. Cannot be set with 'month_frequency' or 'day_of_month'.
month_frequency***	How often, in months, to charge the customer. Cannot be set with 'day_frequency'. Must be set with 'day_of_month'. Values: 1 through 24
day_of_month***	The day that the customer will be charged. Cannot be set with 'day_frequency'. Must be set with 'month_frequency'. Values: 1 through 31 - for months without 29, 30, or 31 days, the charge will be on the last day
start_date	The first day that the customer will be charged. Format: YYYYMMDD
ccnumber****	Credit card number.

ccexp****	Credit card expiration. Format: MMY
payment†	The type of payment. Default: 'creditcard' Values: 'creditcard' or 'check'
checkname†	The name on the customer's ACH account.
checkaccount†	The customer's bank account number.
checkaba†	The customer's bank routing number.
account_type†	The customer's ACH account type. Values: 'checking' or 'savings'
account_holder_type†	The customer's ACH account entity. Values: 'personal' or 'business'
sec_code	ACH standard entry class codes. Values: 'PPD', 'WEB', 'TEL', or 'CCD'
first_name	Cardholder's first name.
last_name	Cardholder's last name.
address1	Card billing address.
city	Card billing city
state	Card billing state.
zip	Card billing postal code.
country	Card billing country code.
phone	Billing phone number.
email	Billing email address.
company	Cardholder's company.
address2	Card billing address, line 2.
fax	Billing fax number.
orderid	Order ID
order_description	Order Description
merchant_defined_field_#	Can be set up in merchant control panel under 'Settings'-&gt;'Merchant Defined Fields'.
ponumber	Cardholder's purchase order number.
processor_id	If using Multiple MIDs, route to this processor (processor_id is obtained under Settings-&gt;Load Balancing in the Control Panel).
customer_receipt	If set to true, when the customer is charged, they will be sent a transaction receipt. Values: 'true' or 'false'

\* Always required

\*\* Required unless 'month\_frequency' and 'day\_of\_month' is set.

\*\*\* Required unless 'day\_frequency' is set.

\*\*\*\* Required for credit card transactions

† Required for ACH transactions

## Update a Subscription's Billing Information

Variable Name	Description
recurring*	Update the subscription's billing information. Value: 'update_subscription'
subscription_id*	The subscription ID that will be updated.

\* Always required

## Delete a Subscription

Variable Name	Description
recurring*	Delete the subscription. Customer will no longer be charged. Value: 'delete_subscription'
subscription_id*	The subscription ID that will be deleted.

\* Always required

## Customer Vault Variables

### Direct Post API

#### Add/Update Customer Record

Variables	Description
customer_vault*	Add/Update a secure customer vault record. Values: 'add_customer' or 'update_customer'
customer_vault_id	Specifies a customer vault id. If not set, the payment gateway will randomly generate a customer vault id.
billing_id	Billing id to be assigned or updated. If none is provided, one will be created or the billing id with priority '1' will be updated.
username*	Username assigned to merchant account.
password*	Password assigned to merchant account.
ccnumber**	Credit card number.
ccexp**	Credit card expiration. Format: MMY
checkname***	The name on the customer's ACH account.
checkaba***	The customer's bank routing number.
checkaccount***	The customer's bank account number.
account_holder_type***	The customer's ACH account entity. Values: 'personal' or 'business'
account_type***	The customer's ACH account type. Values: 'checking' or 'savings'

sec_code	ACH standard entry class codes. Values: 'PPD', 'WEB', 'TEL', or 'CCD'
currency	Set transaction currency.
payment	Set payment type to ACH or credit card. Values: 'creditcard' or 'check'
orderid	Order id
order_description	Order Description
merchant_defined_field_#	Can be set up in merchant control panel under 'Settings'->'Merchant Defined Fields'. Format: merchant_defined_field_1=Value
ponumber****	Cardholder's purchase order number.
tax****	Total tax amount. Format: x.xx
tax_exempt****	Set whether cardholder is tax exempt. Values: 'true' or 'false'
shipping****	Total shipping amount
first_name	Cardholder's first name.
last_name	Cardholder's last name.
address1	Card billing address.
city	Card billing city
state	Card billing state.
zip	Card billing postal code.
country	Card billing country code.
phone	Billing phone number.
email	Billing email address.
company	Cardholder's company.
address2	Card billing address, line 2.
fax	Billing fax number.
shipping_id	Shipping entry id. If none is provided, one will be created or the billing id with priority '1' will be updated.
shipping_firstname	Shipping first name.
shipping_lastname	Shipping last name.
shipping_company	Shipping company.
shipping_address1	Shipping address.
shipping_address2	Shipping address, line 2.
shipping_city	Shipping city
shipping_state	Shipping state.
shipping_zip	Shipping postal code.
shipping_country	Shipping country code.
shipping_phone	Shipping phone number.
shipping_fax	Shipping fax number.
shipping_email	Shipping email address.

- \* Always required
- \*\* Required for credit card transactions
- \*\*\* Required for ACH transactions
- \*\*\*\* Required for Level 2 transactions

## Customer Vault initiated Sale/Auth/Credit/Offline

Variable	Description
username*	Username assigned to merchant account.
password*	Password assigned to merchant account.
customer_vault_id*	Specifies a customer vault id.
amount	Total amount to be charged. For validate, the amount must be omitted or set to 0.00. Format: x.xx
currency	The transaction currency. Format: ISO 4217
processor_id	If using Multiple MIDs, route to this processor (processor_id is obtained under Settings->Load Balancing in the Control Panel).
descriptor	Set payment descriptor on supported processors.
descriptor_phone	Set payment descriptor phone on supported processors.
orderdescription	Order description.
orderid	Order ID

\* Always required

## Delete Customer Record

Variable	Description
customer_vault*	Deletes a secure customer vault record. Values: 'delete_customer'
customer_vault_id*	Specifies a customer vault id.
username*	Username assigned to merchant account.
password*	Password assigned to merchant account.

\* Always required

### Notes:

- If you do not pass a customer\_vault\_id, our system will randomly generate one. If you include a customer\_id and customer\_vault\_id, they must match.
- You can only pass Credit Card **or** Electronic Check transaction variables.

## Partial Payment Information

### Direct Post API

## Request Details

Variable	Description
partial_payments_id	Unique identifier returned when making the original transaction. This should only be used for secondary transactions.
partial_payments	This variable allows the following two values to be passed to it: settle_partial: Settles any amount of tender collected (captured partial auth's and approved partial sales) at cut off. payment_in_full: Required that any split tendered transaction is collected in-full before settlement gets initiated.
type	This variable can be passed the value 'complete_partial_payment' which will complete a payment_in_full transaction that has not been collected in full. This allows industries that require payment_in_full but subsequently decide to still settle the transaction even though it has not been collected in full.

## Response Details

Variable	Description
partial_payment_id	A numeric identifier which is used when submitting subsequent transactions.
partial_payment_balance	Returns the payment's remaining balance.
amount_authorized	Provides the amount that was authorized.

## Examples

Example 1: In this request, if nothing more was done, a transaction for 30.00 would settle at the next cut-off.

Request ...type=sale&partial\_payments=settle\_partial&ccnumber=4111111111111111&ccexp=1016&amount=100.00

Response ...response=1&partial\_payment\_id=123456789&partial\_payment\_balance=70.00&amount=30.00...

Example 2: In this request, payment\_in\_full was required and two transaction were collected - this transaction would settle at the next cut-off.

Request 1 ...type=sale&partial\_payments=payment\_in\_full&ccnumber=4111111111111111&ccexp=1016&amount=100.00

Response 1 ...response=1&partial\_payment\_id=123456789&partial\_payment\_balance=70.00&amount=30.00...

Request 2 ...type=sale&partial\_payment\_id=123456789&partial\_payments=payment\_in\_full&ccnumber=400000000000

Response 2 ...response=1&partial\_payment\_id=123456789&partial\_payment\_balance=0.00&amount=70.00...

Example 3: In this example, payment\_in\_full was required and two transactions were attempted, but only one collected. The merchant decided to force it out anyways - this transaction would settle at the next cut-off.

Request 1 ...type=sale&partial\_payments=payment\_in\_full&ccnumber=4111111111111111&ccexp=1016&amount=100.00

Response 1 ...response=1&partial\_payment\_id=123456789&partial\_payment\_balance=70.00&amount=30.00...

Request 2 ...type=sale&partial\_payment\_id=123456789&partial\_payments=payment\_in\_full&ccnumber=400000000000

Response 2 ...response=2&partial\_payment\_id=123456789&partial\_payment\_balance=70.00&amount=70.00...

Request 3 ...type=complete\_partial\_payment& partial\_payment\_id=123456789&partial\_payments=payment\_in\_full&a

Response 3 ...response=1& partial\_payment\_id=123456789&partial\_payment\_balance=0.00&amount=70.00...

# Transaction Response Variables

## Direct Post API

### Standard Response

Variable Name	Description
response	1 = Transaction Approved 2 = Transaction Declined 3 = Error in transaction data or system error
responsetext	Textual response
authcode	Transaction authorization code.
transactionid	Payment gateway transaction id.
avsresponse	AVS response code (See Appendix 1).
cvvresponse	CVV response code (See Appendix 2).
orderid	The original order id passed in the transaction request.
response_code	Numeric mapping of processor responses (See Appendix 3).

# Testing Information

## Direct Post API

### Transaction testing credentials

Transactions can be tested using one of two methods. First, transactions can be submitted to any merchant account that is in test mode. Keep in mind that if an account is in test mode, all valid credit cards will be approved but **no charges will actually be processed.**

The Payment Gateway demo account can also be used for testing at any time. Please use the following username and password for testing with this account:

username: demo  
password: password

## Transaction POST URL

Transaction details should be POST'ed to the following URL:

POST URL: <https://alliedsynergygroup.transactiongateway.com/api/transact.php>

## Test Data

Transactions can be submitted using the following information:

Visa: 4111111111111111  
MasterCard: 5431111111111111  
Discover: 6011601160116611  
American Express: 3411111111111111  
Credit Card Expiration: 10/25  
account (ACH): 123123123  
routing (ACH): 123123123

## Triggering Errors in Test Mode

- To cause a declined message, pass an amount less than 1.00.
- To trigger a fatal error message, pass an invalid card number.
- To simulate an AVS match, pass 888 in the address1 field, 77777 for zip.
- To simulate a CVV match, pass 999 in the cvv field.

# Appendix 1

## Direct Post API

### AVS Response Codes

X Exact match, 9-character numeric ZIP  
Y Exact match, 5-character numeric ZIP  
D Exact match, 5-character numeric ZIP  
M Exact match, 5-character numeric ZIP  
A Address match only  
B Address match only  
W 9-character numeric ZIP match only  
Z 5-character ZIP match only

- P 5-character ZIP match only
- L 5-character ZIP match only
- N No address or ZIP match only
- C No address or ZIP match only
- U Address unavailable
- G Non-U.S. issuer does not participate
- I Non-U.S. issuer does not participate
- R Issuer system unavailable
- E Not a mail/phone order
- S Service not supported
- O AVS not available
- B AVS not available

## **Appendix 2**

### **Direct Post API**

#### **CVV Response Codes**

- M CVV2/CVC2 match
- N CVV2/CVC2 no match
- P Not processed
- S Merchant has indicated that CVV2/CVC2 is not present on card
- U Issuer is not certified and/or has not provided Visa encryption keys

## **Appendix 3**

### **Direct Post API**

#### **Result Code Table**

- 100 Transaction was approved.
- 200 Transaction was declined by processor.
- 201 Do not honor.
- 202 Insufficient funds.
- 203 Over limit.
- 204 Transaction not allowed.
- 220 Incorrect payment information.
- 221 No such card issuer.
- 222 No card number on file with issuer.
- 223 Expired card.

224 Invalid expiration date.  
225 Invalid card security code.  
240 Call issuer for further information.  
250 Pick up card.  
251 Lost card.  
252 Stolen card.  
253 Fraudulent card.  
260 Declined with further instructions available. (See response text)  
261 Declined-Stop all recurring payments.  
262 Declined-Stop this recurring program.  
263 Declined-Update cardholder data available.  
264 Declined-Retry in a few days.  
300 Transaction was rejected by gateway.  
400 Transaction error returned by processor.  
410 Invalid merchant configuration.  
411 Merchant account is inactive.  
420 Communication error.  
421 Communication error with issuer.  
430 Duplicate transaction at processor.  
440 Processor format error.  
441 Invalid transaction information.  
460 Processor feature not available.  
461 Unsupported card type.