



## **API Documentation**

February 2006

**Network Merchants, Inc.  
Elgin, IL 60123**

**847.352.4850  
support@NetworkMerchants.com**

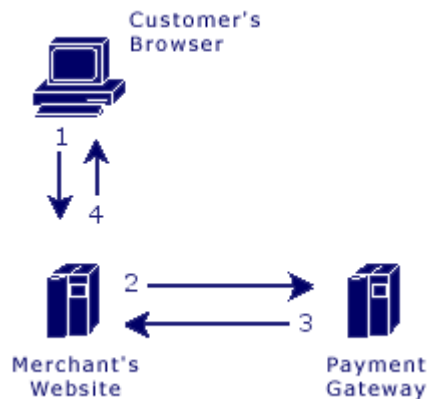
# Table of Contents

<b>Methodology</b> .....	<b>2</b>
Direct Post Method ( <i>Server to Server</i> ).....	2
Browser Redirect Method ( <i>Browser to Server</i> ) .....	3
<b>Transaction Types</b> .....	<b>4</b>
Sale (sale).....	4
Authorization (auth).....	4
Capture (capture) .....	4
Void (void).....	4
Refund (refund).....	4
Credit (credit).....	4
<b>Transaction POST URL</b> .....	<b>5</b>
<b>Transaction Variables</b> .....	<b>5</b>
Sale/Authorization/Credit .....	5
Capture.....	6
Void.....	6
Refund.....	6
Browser Redirect Specific Transaction Variables .....	6
<b>Transaction Response Variables</b> .....	<b>7</b>
Standard Response .....	7
<b>Testing Information</b> .....	<b>8</b>
Transaction Testing Account .....	8
Test Transaction Information.....	8
Triggering Errors in Test Mode .....	8
<b>Examples</b> .....	<b>9</b>
Direct Post.....	9
Browser Redirect .....	9
<b>Changelog</b> .....	<b>10</b>
<b>Appendix 1 – AVS Response Codes</b> .....	<b>10</b>
<b>Appendix 2 – CVV Response Codes</b> .....	<b>10</b>

## Methodology

There are two primary options in which transactions can be submitted through the Payment Gateway API. The most direct and transparent method is our direct post method.

### Direct Post Method (Server to Server)



1. The customer sends their payment information to the merchant's web site.
2. The merchant's 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's web site displays the appropriate message to the customer.

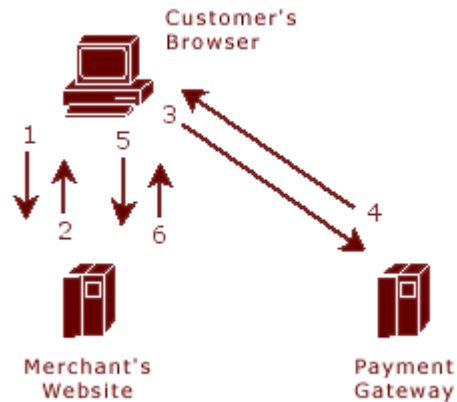
The communication method used to send messages to the Payment Gateway's server is the standard HTTP protocol over an SSL connection.

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`

## Browser Redirect Method (*Browser to Server*)



1. The customer begins the merchant's checkout process.
2. The merchant returns a payment page to the customer with the form's action pointing to the Payment Gateway.
3. The customer **posts** payment data to the Payment Gateway by submitting the form.
4. The Payment Gateway processes the transaction and **redirects** the customer back to the merchant's website with the appropriate results in the GET query string.
5. Because of the redirect, the customer's browser automatically requests the previously passed redirect page on the merchant's web site.
6. The merchant's web site interprets the query string variables and displays the appropriate message to the customer.

*Please note that the merchant may be required to use the **Browser Redirect** method if they are utilizing specific cardholder authentication functionality.*

The redirect method is a bit more involved because the cardholder will be submitting payment data directly to the Payment Gateway but the response must be relayed back to the merchant's web site.

Upon the cardholder's request to checkout (**Step 1**), the merchant must display a form to the cardholder (**Step 2**) that contains or collects the appropriately named variables (as defined below). The additional redirect variable (as defined below) will indicate to the Payment Gateway where the cardholder should be directed and essentially what file will parse the transaction results. This will typically be passed as a hidden field. This form should also have an action of the Payment Gateway's URL.

When the cardholder submits the form, the data is posted to the Payment Gateway (**Step 3**). The Payment Gateway processes the transaction and sends a header redirect back to the cardholder (**Step 4**).

In **Step 5**, the cardholder requests the URL defined by the previously posted redirect variable and the results of the transaction are included in the GET query string. The merchant must create a script that parses these response variables, updates their database/ordering system appropriately, and displays the receipt or decline message to the cardholder. (**Step 6**)

## Transaction Types

### Sale (sale)

Transaction sales are submitted and **immediately flagged for settlement**. These transactions will automatically be settled.

### Authorization (auth)

Transaction authorizations are authorized immediately but are **not flagged for settlement**. These transactions must be flagged for settlement manually using the *capture* transaction type. Authorizations typically remain activate 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 a negative amount to the cardholder's card. In most situations credits are disabled as transaction refunds should be used instead.

## Transaction POST URL

All test and live transactions should be submitted to the following url:

**<https://secure.networkmerchants.com/gw/api/transact.php>**

This URL is the same for both Direct Post and Browser Redirect methods.

## Transaction Variables

### Sale/Authorization/Credit

Variable Name	Required*	Format	Description
type	<b>Required</b>	sale / auth / credit	sale = Transaction Sale auth = Transaction Auth credit = Transaction Credit
username	<b>Required</b>		Username assigned to merchant account
ccnumber	<b>Required</b>		Credit card number
ccexp	<b>Required</b>	MMYY	Credit card expiration (ie. 0705 = 7/2005)
amount	<b>Required</b>	x.xx	Total amount to be charged (i.e. 10.00)
cvv	Recommended		Card security code
orderid	Recommended		Order ID
orderdescription	<i>Optional</i>		Order description
ipaddress	Recommended	xxx.xxx.xxx.xxx	IP address of the cardholder
tax	<i>Level II</i>	x.xx	Total tax amount
shipping	<i>Level II</i>	x.xx	Total shipping amount
ponumber	<i>Level II</i>		Original Purchase Order
firstname	Recommended		Cardholder's first name
lastname	Recommended		Cardholder's last name
company	<i>Optional</i>		Cardholder's company
address1	Recommended		Card billing address
address2	<i>Optional</i>		Card billing address – line 2
city	Recommended		Card billing city
state	Recommended	CC	Card billing state (2 character abbrev.)
zip	Recommended		Card billing zip code
country	Recommended	CC (ISO-3166)	Card billing country (ie. US)
phone	Recommended		Billing phone number
fax	<i>Optional</i>		Billing fax number
email	Recommended		Billing email address
website	<i>Optional</i>		Website
shipping_firstname	<i>Optional</i>		Shipping first name
shipping_lastname	<i>Optional</i>		Shipping last name
shipping_company	<i>Optional</i>		Shipping company
shipping_address1	<i>Optional</i>		Shipping address
shipping_address2	<i>Optional</i>		Shipping address – line 2
shipping_city	<i>Optional</i>		Shipping city
shipping_state	<i>Optional</i>		Shipping state
shipping_zip	<i>Optional</i>		Shipping zip code
shipping_country	<i>Optional</i>	CC (ISO-3166)	Shipping country (ie. US)
shipping_email	<i>Optional</i>		Shipping email address

*\*These fields are required by default. Level II fields are required for Level II processing. Recommended fields help provide additional address and cardholder verification. Certain banks may require some optional fields.*

## Capture

Variable Name	Required	Format	Description
type	Required	capture	capture = Transaction Capture
username	Required		Username assigned to merchant account
transactionid	Required		Original Network Merchants transaction id
amount	Required	x.xx	Total amount to be settled (i.e. 10.00) <i>This amount must be equal to or less than the original authorized amount.</i>

## Void

Variable Name	Required	Format	Description
type	Required	void	void = Transaction Capture
username	Required		Username assigned to merchant account
password	Required		Password for the specified username
transactionid	Required		Original Network Merchants transaction id

## Refund

Variable Name	Required	Format	Description
type	Required	refund	refund = Transaction Capture
username	Required		Username assigned to merchant account
password	Required		Password for the specified username
transactionid	Required		Original Network Merchants transaction id
amount	Required	x.xx	Total amount to be refunded (i.e. 10.00) <i>This amount must be equal to or less than the settled amount.</i>

## Browser Redirect Specific Transaction Variables

All of the above variables pertain to the browser redirect; however, the variables listed below are also part of this method and are not included in the Direct Post method.

Variable Name	Required	Format	Description
redirect	Required for Browser Redirect	https://...	The URL the cardholder should be redirected to. This URL must also parse and respond to the response variables included in the GET query upon the redirect.

# Transaction Response Variables

## Standard Response

Variable Name	Format	Description
response	1 / 2 / 3	1 = Transaction Accepted 2 = Transaction Declined 3 = Error in transaction data or system error
responsetext		Textual response
authcode		Transaction authorization code
transactionid		Network Merchants transaction id
avsresponse	C	AVS Response Code (See Appendix 1)
cvvresponse	C	CVV Response Code (See Appendix 2)
orderid		The original order id passed in the transaction request.
<b>Browser Redirect Specific Response Variables</b>		
username		The original username passed in the transaction request.
time		The time of the response in seconds since Epoch (Midnight UTC Jan 1, 1970 )
amount		The original amount passed in the transaction request.
key		An MD5 hash of the following variables <b>pipe</b> delimited. This is used to verify the authenticity of the response:  username password orderid response transactionid amount time avsresponse cvvresponse

## Testing Information

### Transaction Testing Account

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 Network Merchants 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

### Test Transaction Information

Test transactions should be submitted with the following information:

**Credit Card Number:** 4111111111111111  
**Credit Card Expiration:** Any valid expiration date  
**Amount:** >1.00

### Triggering Errors in Test Mode

To cause a declined message, pass an amount less than 1.00.

To trigger an error message, pass an invalid card number.

# Examples

## Direct Post

### Data posted by merchant's web site to Payment Gateway

```
username=demo&password=password&type=sale&ccnumber=4444444444444444&ccexp=0110&cvv=123&amount=10.00
```

### Response data returned to merchant's web site in HTML body

```
response=3&responsetext=Invalid+Card&authcode=&transactionid=12345&avsresponse=&cvvresponse=
```

## Browser Redirect

### Form generated by merchant and ultimately posted to the Payment Gateway by cardholder

```
<form method="post" action="https://secure.../transact.php">  
<input type=hidden name=username value=demo>  
<input type=hidden name=ccnumber value=4444444444444444>  
<input type=hidden name=ccexp value=0110>  
<input type=hidden name=amount value=10.00>  
<input type=hidden name=redirect value="https://example.com/res.cgi">  
</form>
```

### Upon completion (approved/declined/error), cardholder is redirected to:

```
https://example.com/res.cgi?response=3&responsetext=Invalid+Card&authcode=&transactionid=12345&avsresponse=&cvvresponse=&orderid=&username=demo&amount=10.00&time=1139861680&key=5f961e415e1bfb69c2c515f91cb17b16
```

Note that 5f961e415e1bfb69c2c515f91cb17b16 is the md5 hash of:

```
demo|password||3|12345|10.00|1139861680||
```

# Changelog

February 2006  
November 2004

Documented Browser Redirect Capability  
PDF API documentation released.

## Appendix 1 – AVS Response Codes

X	Exact match, 9-character numeric ZIP
Y	Exact match, 5-character numeric ZIP
D	“
M	“
A	Address match only
B	“
W	9-character numeric ZIP match only
Z	5-character Zip match only
P	“
L	“
N	No address or ZIP match
C	“
U	Address unavailable
G	Non-U.S. Issuer does not participate
I	“
R	Issuer system unavailable
E	Not a mail/phone order
S	Service not supported
0	AVS Not Available
O	“
B	“

## Appendix 2 – 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