IT Information Services

API Developer's Guide

Version 6.0

August 12, 2007

TABLE OF CONTENTS

TABLE OF TABLES

1 Purpose

The purpose of this document is to provide technical guidelines for integrating to and using the VerePay credit card gateway.

This documentation is intended to be used by software developers comfortable with the use of standard web communication techniques including HTTPS Post or Get and XML Post methods.

2 Overview

2.1 VerePay Gateway Switch

The VerePay Gateway Switch (VerePay) is a gateway that supports the secure transmission of transactions through the use of a simple-to-integrate API.

The gateway is connected to a complex rules-based switch that routes the transaction to the appropriate credit card processor. The switch is able to route a transaction based on the terminal number, card type as well as the transaction type making it easier to switch a merchant from one processor to another.

2.2 Technical Services

Once VerePay has been implemented into your payment system, no more updates are required unless you add features as they become available. Any other modifications required are handled by our office.

2.2.1 Switching Processors

If a merchant wants to switch to a new processor we point them to the new processor either for all transactions or just the card types they want to switch. We then ensure that settlement, void and refund of transactions are sent to the original processor for the authorization while new transactions are routed to the new processor.

2.2.2 Online Portal Application

VerePay includes an online Portal application that is accessed through a Web browser. This portal includes a powerful virtual terminal and several reports that aid the merchant in researching transactions. The Portal application also allows authorizations to be settled or voided and settled transactions to be refunded.

2.2.2.1 Fraud Protection

VerePay provides several measures to combat cardholder fraud. From the portal a credit card can be blocked preventing the card from being used on the merchant's Web site. The portal provides flexibility allowing the user to block a fraudulent card.

As an additional fraud prevention tool, we have set the allowable the AVS and CVV responses at the terminal level while providing the merchant the greatest acceptance flexibility for the merchant type. Both domestic and international AVS and CVV are supported

2.2.2.2 Security Information

VerePay utilizes HTTPS Secure Socket Layers for both the API and the Portal application. VerePay resides in a state of the art processing facility that is PCI certified. Load balancing and redundancy are utilized to provide for maximum availability. The architecture of the application makes it impossible to access our database server from the web. All sensitive data is encrypted including card numbers, terminal numbers, and all credentials using 3DES using multiple keys and vectors, for extra security. In compliance with PCI requirements, complete card numbers are not available on any of our systems.

3 Testing and Production Environments

Separate testing and production environments support the testing of new versions of solutions without affecting the existing production version. Dedicated support staff is always on hand to help you during your integration via email and telephone.

3.1 Test System

The Name/Value pair API is located at http://dev.verepay.cc/ezpay/ipay.aspx.

The test system is outside of our PCI environment and is not secured by SSL. For this reason we do not recommend the use of live card numbers. Instead, use the test card numbers shown below for the test system.

Card Type	Test Card Number
Visa	4123456789012349
MasterCard	5123456789012346

The test system Portal is located at https://dev.verepay.cc. Before you can begin testing you will need to obtain a user id and password for the test portal.

3.2 Obtaining a User ID and Password for the Test Portal

Log on to the test portal to retrieve the terminal number and store key required to submit transactions. E-mail support@verpay.com to request a test logon. Be sure to include your name, company name, and contact information.

3.3 Production System

The Production Name/Value pair API is located at https://www.verepay.cc/ezpay/ipay.aspx.

The Production Portal is located at https://www.verepay.cc.

Processor credentials are safeguarded by requiring unique credentials for each terminal within the switch. This also makes it easier to switch processors as the processor credentials are never sent in with the transaction request.

VerePay credentials include a terminal number and terminal key that are sent in with every transaction request as the terminal and Store Key parameters.

Credentials are retrieved from the Portal and are located under the Account Settings menu item and Credential Review sub item. The Credential Review page is also used to generate a new terminal key. It is recommended to generate a new key once a month. If you believe the terminal key has been compromised, generate a new key immediately.

> **Note**
> *For more details regarding the use of the Portal, see the Portal Merchant User Guide.*

3.4 Transaction Types

VerePay supports the following transaction types.

- Authorization
- Capture of a previous Authorization (this marks the transaction for settlement)
- Void of a previous Authorization
- Refund of a previously settled transaction
- Test (this ensures the system is responding)

Blind credits are not supported by the API but can be entered through the Virtual Terminal of the Portal.

4 API

4.1 Overview

VerePay allows transactions to be submitted through our API providing rapid integration of VerePay into your payment solution. It supports HTTPS Get, HTTPS Post, and XML Post request methods.

Code examples for how to send and receive transactions using these methods can be found in Appendix C – Code Samples on page 19.

4.2 Transaction Request and Response Parameters

The following tables display all parameters that can be passed as part of the supported transaction types in the transaction request, as well as the parameters that will be received back with the response.

4.2.1 Authorization

An authorization transaction submits a credit card authorization to the VerePay switch. If the terminal is set up to automatically settle all successful authorizations, an authorization transaction will also capture the authorization after it has been authorized. If the terminal is not set up to automatically settle successful authorizations, transactions must be settled manually either by sending a Capture of a Previous Authorization transaction through the API or by using the Virtual Terminal or Transaction Review to settle the transaction through the Portal.

4.2.1.1 Request

The following table displays parameters that make up an authorization transaction request.

Field Name	Required	Description
type	Yes	The type of transaction being performed **Auth or A**
terminal	Yes	The VerePay terminal number which can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**
storeKey	Yes	The VerePay terminal key that can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**
vendor	Yes	Internal code used by VerePay **bcc**
cardNumber	Yes	Credit Card Number
expiryDate	Yes	Expiration date of card. This must be 4 characters in the format mmyy
cvvData	No	The 3 or 4 digit security code from the card
tranAmount	Yes	Authorization amount. All amounts are assumed to include 2 decimal places. The decimal point is optional. (For example, a tranAmount sent in as 100 will run as 1.00)

Field Name	Required	Description
taxAmount	No	Total amount of tax
shippingAmount	No	Total shipping cost
billName	No	Name on the card
billAddress	No	First line of the card's billing address
billCity	No	City of the card's billing address
billState	No	State of the card's billing address
billZipCode	No	Zip code of the card's billing address
billCountry	No	Country of the card's billing address
billPhone	No	Phone number of the card's billing address
billEmail	No	Email address of the card's billing address
shipName	No	Customer Name of the shipping address
shipAddress	No	Address of the shipping address
shipCity	No	City of the shipping address
shipState	No	State of the shipping address
shipZipCode	No	Zip Code of the shipping address
shipCountry	No	Country of the shipping address
shipPhone	No	Phone number of the shipping address
shipEmail	No	E-mail address of the shipping address
clientIP	No	IP address of the purchaser
invoiceNumber	No	Merchant order number for the transaction
invoiceDescription	No	Short description of the transaction
localDate	No	Transaction Date in the format yyyymmdd
localTime	No	Transaction Time in the format hhmmss
supplierID	No	Optional Supplier ID
affiliateID	No	Optional Affiliate ID
eci	No	ECI returned by PaymentAuthentication
mares	No	MARES returned by PaymentAuthentication. This must be URL encoded
vbvEnrolled	No	Y/N value returned by PaymentAuthentication
vbvVerified	No	Y/N value returned by PaymentAuthentication

Table 1: Authorization Transaction Request

4.2.1.2 Response

The response for an authorization transaction will always contain a result parameter. Always test the result first to see if the transaction was successful. A zero (0) value means the authorization was successful. The following table displays all the parameters that make up an authorization response.

Field Name	Description
result	Result of the transaction. A value of 0 denotes that the transaction was successful. A value of 1 denotes that the transaction failed. **0 or 1**
tranID	Unique identifier tracked throughout the transaction lifecycle.

Field Name	Description
	Required to capture, void, or refund a transaction.
largeTranID	12-digit unique identifier used by merchants. Same as the tranID with a one and zeros added.
authCode	Authorization Code for the transaction
authMessage	Message showing the result of the transaction. This will either be a message generated by the processor or will be a message generated by the VerePay switch. See Appendix A – VerePay Response Messages.
avsResponse	Result of AVS. Address verification.
cvvResponse	Card Security Code validation result
tranDate	Date the transaction took place. Required format yyyymmdd.
tranTime	Time the transaction took place. Required format hhmmss.

Table 2: Authorization Transaction response

4.2.2 Capture a Previous Authorization

A Capture of a Previous Authorization transaction submits a capture to the VerePay switch. This will mark the authorization for settlement. Depending on the processor that is servicing the account, the settlement will either be sent to the processor immediately, or will be marked for batch settlement and then sent to the processor on a schedule.

4.2.2.1 Request

The following table displays the parameters that make up a capture of a previous authorization transaction request.

Field Name	Required	Description
type	Yes	The type of transaction being performed **Capture or S**
terminal	Yes	The VerePay terminal number that can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**
storeKey	Yes	The VerePay terminal key that can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**
vendor	Yes	Internal code used by VerePay **bcc**
origTranID	Yes	Original transaction id for the authorization that is to be captured. If the original authorization was submitted through the VerePay API, this relates to the tranID parameter from the response.
origTranAmount	Yes	Original transaction amount for the authorization that is to be captured. All amounts are assumed to include 2 decimal places. The decimal point is optional.
tranAmount	Yes	Transaction amount to be captured. All amounts are assumed to include 2 decimal places. The decimal point is optional.
taxAmount	No	Tax amount.

Field Name	Required	Description
localDate	No	Transaction Date in the format yyyymmdd
localTime	No	Transaction Time in the format hhmmss
clientIP	No	IP address of the customer

Table 3: Capture Previous Authorization Request

4.2.2.2 Response

The response for a Capture of a Previous Authorization transaction will always contain a result parameter. Always test the result first to see if the transaction was successful. A zero (0) value means the capture was successful. The following table displays all the parameters that ma ke up a Capture of a Previous Authorization response.

Field Name	Description
result	Result of the transaction. A value of 0 denotes that the transaction was successful. A value of 1 denotes that the transaction failed. **0 or 1**
tranID	Unique identifier tracked throughout the transaction lifecycle. Required to refund the transaction.
largeTranID	12-digit unique identifier used by merchants. Same as the tranID with a one and zeros added.
authCode	Authorization Code for the transaction
authMessage	Message showing the result of the transaction. This will either be a message generated by the processor or will be a message generated by the VerePay switch. See Appendix A for VerePay Messages.
avsResponse	Does not contain a value for a Capture of a Previous Authorization.
cvvResponse	Does not contain a value for a Capture of a Previous Authorization.
tranDate	Date the transaction took place. Required format yyyymmdd.
tranTime	Time the transaction took place. Required format hhmmss.

Table 4: Capture Previous Authorization Response

4.2.3 Void of a Previous Authorization

A Void of a Previous Authorization transaction submits a void to the VerePay switch. This will mark the authorization as Voided. This will prevent the authorization from being settled or refunded.

4.2.3.1 Request

The following table displays the parameters that make up a void of a previous authorization transaction request.

Field Name	Required	Description
type	Yes	The type of transaction being performed **Void or V**
terminal	Yes	The VerePay terminal number which can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**
storeKey	Yes	The VerePay terminal key which can be retrieved from the Credential Review page of the VerePay Portal **Retrieved From Portal**

Field Name	Required	Description
vendor	Yes	Internal code used by VerePay **bcc**
origTranID	Yes	Original transaction id for the authorization that is to be voided. If the original authorization was submitted through the VerePay API, this relates to the TranID parameter from the response.
origTranAmount	Yes	Original transaction amount for the authorization that is to be voided. All amounts are assumed to include 2 decimal places. The decimal point is option.
localDate	No	Transaction Date in the format yyyymmdd
localTime	No	Transaction Time in the format hhmmss
clientIP	No	IP address of the customer

Table 5: Void of Previous Authorization Transaction Request

4.2.3.2 Response

The response for a Void of a Previous Authorization transaction will always contain a result parameter. Always test the result first to see if the transaction was successful. A zero (0) value means the void was successful. The following table displays all the parameters that make up a Void of a Previous Authorization response.

Field Name	Description
result	Result of the transaction. A value of 0 denotes that the transaction was successful. A value of 1 denotes that the transaction failed. **0 or 1**
tranID	Transaction ID for the transaction.
largeTranID	12-digit unique identifier used by merchants. Same as the tranID with a one and zeros added.
authCode	Authorization Code for the transaction
authMessage	Message showing the result of the transaction. This will either be a message generated by the processor or will be a message generated by the VerePay switch. See Appendix A for VerePay Messages.
avsResponse	Does not contain a value for a Void of a Previous Authorization
cvvResponse	Does not contain a value for a Void of a Previous Authorization
tranDate	Date the transaction took place. Required format is yyyymmdd.
tranTime	Transaction Time. The format of the transaction time is hhmmss.

Table 6: Table 7: Void of Previous Authorization Transaction Response

4.2.4 Refund of a Previously Settled Transaction

A Refund of a Previously Settled Transaction submits a refund to the VerePay switch. This will issue a credit against the original transaction. Partial refunds are allowable. Multiple refunds can be sent against a single settled transaction as long as the total amount refunded does not exceed the original settled transaction amount.

4.2.4.1 Request

The following table displays the parameters that make up a refund of a previously settled transaction request.

Field Name	Required	Description

Field Name	Required	Description
type	Yes	The type of transaction being performed. **Refund or R**
terminal	Yes	The VerePay terminal number that can be retrieved from the Credential Review page of the VerePay Portal. **Retrieved From Portal**
storeKey	Yes	The VerePay terminal key that can be retrieved from the Credential Review page of the VerePay Portal. **Retrieved From Portal**
vendor	Yes	Internal code used by VerePay **bcc**
localDate	No	Time the transaction took place. Required format yyyymmdd.
localTime	No	Time the transaction took place. Required format hhmmss.
origTranID	Yes	Original transaction id for the authorization that is to be captured.
origTranAmount	Yes	Original transaction amount for the authorization that is to be captured.
tranAmount	Yes	Transaction amount to be captured. All amounts are assumed to include 2 decimal places. The decimal point is optional.
taxAmount	No	Optional tax amount.
clientIP	No	IP address of the customer

Table 8: Previously Settled Transaction Request

4.2.4.2 Response

The response for a Refund of a Previously Settled Transaction will always contain a result parameter. Always test the result first to see if the transaction was successful. A zero (0) value means the refund was successful. The following table displays all the parameters that make up a Refund of a Previously Settled Transaction response.

Field Name	Description
result	Result of the transaction. A value of 0 denotes that the transaction was successful. A value of 1 denotes that the transaction failed. **0 or 1**
tranID	Transaction ID for the transaction.
largeTranID	12-digit unique identifier used by merchants. Same as the tranID with a one and zeros added.
authCode	Authorization Code for the transaction
authMessage	Message showing the result of the transaction. This will either be a message generated by the processor or will be a message generated by the VerePay switch. See Appendix A for VerePay Messages.
avsResponse	Does not contain a value for a Refund of a Previously Settled Transaction.
cvvResponse	Does not contain a value for a Refund of a Previously Settled Transaction.
tranDate	Date the transaction took place. Required format is yyyymmdd.

Field Name	Description
tranTime	Time the transaction took place. Required format is hhmmss.

Table 9: Previously Settled Transaction Response

5 Appendix A – VerePay Response Messages

5.1 Overview

The VerePay Gateway Switch will sometimes generate its own response messages for all transactions. These may denote a format error with the transaction, a decline of the transaction as a result of fraud checking or a problem that was encountered while processing the transaction. Processors will otherwise generate the response messages.

5.2 VerePay Response Messages

The following table displays messages that VerePay generates, that are not associated with processor generated messages. These messages will be returned in the authMessage parameter for all transaction types. In addition to these messages, Processor generated messages will normally be returned in the authMessage parameter. The following table displays the importance of the result parameter and authMessage parameter.

Result	Approve/ Decline	Message	Description
0	Approved	Approved and Captured	For a terminal that is set up to automatically settle all successful authorizations, the authorization was approved and then successfully captured for settlement.
0	Approved	Approved but Capture failed. Please Capture manually	For a terminal that is set up to automatically settle all successful authorizations, the authorization was approved but there was a problem capturing the transaction for settlement. It will be necessary to manually settle the transaction either by sending a Capture of a Previous Authorization transaction through the API or by using the Virtual Terminal or Transaction Review to settle the transaction through the Portal. Details of the Virtual Terminal and Transaction Review can be found in the VerePay Portal Merchant User Guide
1	Declined	Decline AVS and CVV	AVS and CVV results were not in the acceptable values defined for the terminal. The VerePay Gateway Switch restricts AVS and CVV results to prevent fraud and protect the merchant. These settings are made on a terminal by terminal basis.
1	Declined	Decline AVS	The AVS result was not in the acceptable values defined for the terminal. The VerePay Gateway Switch restricts AVS and CVV results to prevent fraud and protect the merchant. These settings are made on a terminal by terminal basis.
1	Declined	Decline CVV	The CVV result was not in the acceptable values defined for the terminal. The VerePay Gateway Switch restricts AVS and

Result	Approve/ Decline	Message	Description
			CVV results to prevent fraud and protect the merchant. These settings are made on a terminal by terminal basis.
1	Declined	Blocked Card	The card number has been blocked in the VerePay Gateway Switch from usage at the Merchant site, a Client's set of sites, or system-wide. This helps prevent fraud by identifying cards that have problems and preventing their use.
1	Format	Format Error	An invalid Terminal, Vendor or Store Key was sent in with the transaction. The Terminal and Store Key must match those that are shown on the Credential Review page of the Portal. Vendor must always be bcc.
1	Format	CardNumber Invalid	An invalid Card Number was sent in with the transaction. (The VerePay Gateway Switch always validates the card number prior to submitting the transaction for processing.)
1	Format	Expiry Date Invalid	An invalid expiration date was sent in with the transaction. The expiration date must be in the format of mmyy.
1	Format	Terminal Error	The Terminal and Store Key sent in with the transaction are not valid or no processors are defined for the terminal. The Terminal and Store Key must match those that are shown on the Credential Review page of the Portal. A processor must have been set up prior to being able to run transactions through the API. If the terminal and store key are correct, please contact your account representative.
1	Format	Original Transaction ID Invalid	For a Refund/Void or Capture, the Original Transaction ID sent in with the transaction is not numeric.
1	Format	Original Transaction Not Found	For a Refund/Void or Capture, the Original Transaction ID was not found. The tranID returned with the original authorization (for a Void or Capture) or original capture (for a refund) must be sent.
1	Format	Invalid Original Transaction	For a Refund, the original transaction does not have a settled status. For a void or capture the original transaction does not have an authorized status.
1	Format	Tran Amount Invalid	The Transaction Amount sent in with the transaction is not in the correct format. All amounts have 2 implied decimals so no decimal point is needed.
1	Format	Tax Amount	The Tax Amount sent in with the transaction

Result	Approve/ Decline	Message	Description
		Invalid	is not in the correct format. All amounts have 2 implied decimals so no decimal point is needed.
1	Format	Shipping Amount Invalid	The Shipping Amount sent in with the transaction is not in the correct format. All amounts have 2 implied decimals so no decimal point is needed.
1	Format	Vendor Error	The Vendor code sent in with the transaction is not a valid vendor. It must be bcc.

Table 10: VerePay Response Messages

6 Appendix B – AVS and CVV Return Codes

6.1 Overview

The VerePay Gateway Switch converts processor AVS and CVV return codes to a single generic set of values allowing for simple switching of merchant processors. The merchant is not required to understand the values for all the processors.

6.2 AVS Return Codes

The following tables displays all the possible AVS return codes that will be returned in the avsResponse parameter for an Authorization transaction. For all other transaction types, the avsResponse will be blank.

6.2.1 Domestic

Value	Description
F	Exact Match. 9 digit Match
M	Exact Match. 5 digit Match
A	Address Only
9	9 digit Match
5	5 digit Match
N	No Match
U	Address Unavailable
R	Issuer System Unavailable
X	Ineligible for AVS
S	Service not supported
I	International Card not supported

Table 11: Domestic AVS Return Codes

6.2.2 International

Value	Description
1	Address and Zip Match
2	UK Address and Zip Match
3	Address Matched
4	Zip Matched
6	Address and Zip Verified
7	Good Address and Zip Match
8	Address Not Verified

Table 12: International AVS Return Codes

6.3 CVV Return Codes

The following tables display all the possible CVV return codes that will be returned in the cvvResponse parameter for an Authorization transaction. For all other transaction types, the cvvResponse will be blank.

Value	Description
M	Match
N	No Match
P	Not Processed
S	CVV2 not present on card
U	Issuer not certified

Table 13: CVV Return Codes

7 Appendix C – Code Samples

7.1 Text/XML Using Post Method Request - C#

The following code example shows how to send a transaction request to VerePay using the XML Post Method.

```
//Create a request
string req = @"<VerePay><type>A</type>"
+ @"<cardNumber>4123456789012349</cardNumber>"
+ @"<vendor>bcc</vendor>"
+ @"<terminal>1700000001</terminal>"
+ @"<storeKey>9999-9999-9999-9999</storeKey>"
+ @"<expiryDate>1210</expiryDate>"
+ @"<cvvData>123</cvvData>"
+ @"<billZipCode>12345</billZipCode>"
+ @"<billName>Test</billName>"
+ @"<tranAmount>0.01</tranAmount></VerePay>";

ASCIIEncoding encoding = new ASCIIEncoding();
byte[] data = encoding.GetBytes(req);

HttpWebRequest webReq =
       (HttpWebRequest)WebRequest.Create("http://dev.verepay.cc/ez
       pay/ipay.aspx");
webReq.Method = "POST";
webReq.ContentType = "text/xml charset=ASCII";
webReq.ContentLength = data.Length;

//Read the response
Stream strm = webReq.GetRequestStream();
strm.Write(data, 0, data.Length);

HttpWebResponse webResp = (HttpWebResponse)hwr.GetResponse();
Stream respStrm = webResp.GetResponseStream();
```

7.2 HTTPS Name/Value Pair Using Get Method Request - C#

The following code example shows how to send a transaction request to VerePay using the HTTPS Get Method.

```
//Create a request
string url = @"http://dev.verepay.cc/ezpay/ipay.aspx?type=auth
+ &terminal=1700000001&storeKey=9999-9999-9999-9999&vendor=bcc"
+ &cardNumber=4123456789012349&expiryDate=1210&tranAmount=0.01";

WebRequest webReq = WebRequest.Create(url);

//Read the response
WebResponse webResp = webReq.GetResponse();
Stream respStrm = webResp.GetResponseStream();
```

7.3 HTTPS Name/Value Pair Using Post Method Request - C#

The following code example shows how to send a transaction request to VerePay using the HTTPS Get Method.

```
//Create a request
string req = "type=A&terminal=1700000001"
+ "&storeKey=9999-9999-9999-9999&vendor=bcc"
+ "&cardNumber=4123456789012349&expiryDate=1217&tranAmount=0.01";

ASCIIEncoding encoding = new ASCIIEncoding();
byte[] data = encoding.GetBytes(req);

HttpWebRequest webReq =
       (HttpWebRequest)WebRequest.Create("http://dev.verepay.cc/
       ezpay/ipay.aspx");
webReq.Method = "POST";
webReq.ContentType = "application/x-www-form-urlencoded";
webReq.ContentLength = data.Length;

Stream reqStrm = hwr.GetRequestStream();
strm.Write(data, 0, data.Length);

//Read the response
HttpWebResponse webResp = (HttpWebResponse)hwr.GetResponse();
Stream respStrm = webResp.GetResponseStream();
```

7.4 Parse the response - C#

A typical transaction result is a collection of name/value pairs separated by a comma and will be in the following format:

```
result=0,tranID=417,largeTranID=100000000417,authCode=HO
STOK,authMessage=Approved,avsResponse=M,cvvResponse=P,tr
anDate=20040604,tranTime=215954
```

Each name/value pair in the response is separated by a comma. The following code example shows how to break apart the string.

```
//Parse the result
Byte[] read = new Byte[512];
int bytes = respStrm.Read(read, 0, 512);

string result = "";

while (bytes > 0)
{
       Encoding encode =
              System.Text.Encoding.GetEncoding("utf-8");
       result = result + encode.GetString(read, 0, bytes);
       bytes = respStrm.Read(read, 0, 512);
}

//Read and process the results
string[] response = result.Split(',');
string key;
string valuePart;

NameValueCollection nr = new NameValueCollection();
int i = 0;
foreach (string xx in response)
{
       i = xx.IndexOf("=");
       key = xx.Substring(0, i);
       valuePart = xx.Substring(i + 1);
       nr.Add(key, valuePart);
}
```

```
//Access indiviual parts of the response
nr["result"].ToString();
nr["tranID"].ToString();
nr["largeTranID"].ToString();
nr["authCode"].ToString();
nr["authMessage"].ToString();
nr["avsResponse"].ToString();
nr["cvvResponse"].ToString();
nr["tranDate"].ToString();
nr["tranTime"].ToString();
```

8 Revision Control Table

Date	Author	Description	Version
07/13/07	R. Myers	Initial Release	V 6.0