Moneris MPI
Verified by Visa / MasterCard SecureCode

Transaction DTD

v.1.1.1

Table of Contents

**** PLEASE READ CAREFULLY****

You have a responsibility to send only Visa or MasterCard to the Moneris MPI. Under no circumstances should ANY other card type be sent to the VBV/MCSC MPI.

1. About this Documentation

The eSELECTplus payment gateway supports Verified by Visa (VbV) and MasterCard SecureCode (MCSC) transactions in XML format over the HTTPS protocol. This document contains detailed information on the request and response transaction requirements of eSELECTplus' MPI XML format. When creating custom API's, these requirements must be met in order for VbV/MCSC transactions to be sent to eSELECTplus in the proper format.

To help prevent fraudulent activity on online transactions it is highly recommended that you also implement the eSELECTplus eFraud features which consist of AVS and CVD.

Address Verification Service (AVS) – Verifies the cardholder's billing address information.

Card validation Digit (CVD) – Validates that cardholder has a genuine credit card in their possession during the transaction.

2. Verified by Visa

Verified by Visa (VbV) is a program initiated by Visa. Before approving a transaction eSELECTplus and the Bank that issues the Visa credit cards will attempt to authenticate the cardholder through the use of a password, similar to a debit PIN. When an authentication is attempted the merchant is protected from chargebacks.

3. MasterCard SecureCode

MasterCard SecureCode (MCSC) is a new feature offered by MasterCard. Merchants who have enrolled in this program with Moneris and eSELECTplus will be able to offer their customers added protection against unauthorized credit card use, as well as protect themselves from fraud-related chargebacks. Cardholders that have applied for SecureCode with their issuing bank will be able to use this password similar to a debit PIN number for online transactions with participating online merchants.

4. What is the Process I will need to follow?

You will need to follow these steps.

1. Do the required development as outlined in this document
2. Test your solution in the test environment
3. Activate your store
4. Make the necessary changes to move your solution from the test environment into production as outlined in this document

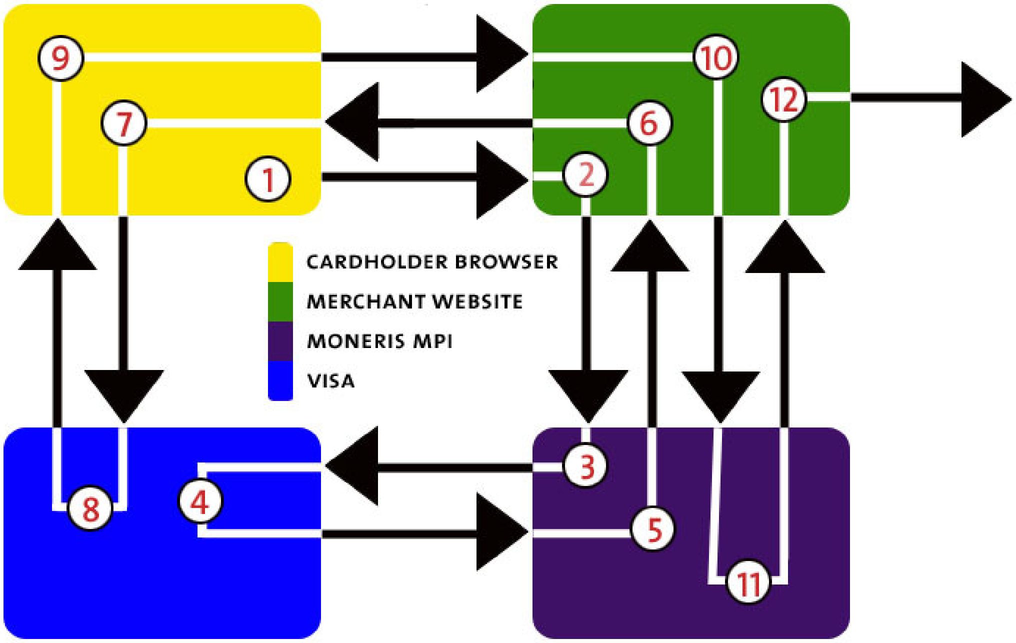
Note:
Your solution may be required to demonstrate compliant with the card associations' PCI/CISP/PABP requirements. For For more information on how to get your application PCI-DSS compliant, please contact our Sales Center and visit https://www.eselectplus.ca/en/downloadable-content to download the PCI_DSS Implementation Guide.

The card association has a couple of data security standards that define specific requirements for all organizations that store, process, or transmit cardholder data. As a Moneris Solutions client or partner using this method of integration, your solution must demonstrate compliance to the Payment Card Industry Data Security Standard (PCI DSS) and/or the Payment Application Data Security Standard (PA DSS). These standards are designed to help the cardholders and merchants in such ways as they ensure credit card numbers are encrypted when transmitted/stored in a database and that merchants have strong access control measures.

Non-compliant solutions may prevent merchant boarding with Moneris Solutions. For further information on PCI DSS & PA DSS requirements, please visit http://www.pcisecuritystandards.org.

5. Transaction Flow

Below is a diagram with explanations about the flow of a VBV/MCSC transaction.



1. Cardholder enters their credit card number and submits their transaction information to the merchant.

2. Upon receiving the transaction request the merchant sends a TXN type request. For sample code please refer to section 9.a.

3. The Moneris MPI receives the request, authenticates the merchant and based on the card type, sends the transaction information to Visa or MasterCard.

4. Visa/MasterCard verifies if the card is enrolled and returns the issuer URL.

5. Moneris MPI receives the response from Visa or MasterCard and forwards the information to the merchant.

6. The merchant receives the response from the eSELECTplus Moneris MPI. If the card is enrolled the response would be "Y" and the merchant generates an inline window in the cardholder's browser. If the response was an "N" (not enrolled), a transaction could be sent to the processor identifying it as VBV/MCSC attempt with a crypt type of 6. If the response was "U" (Unable to Authenticate) or the response times out, the transaction can be sent to the processor with a crypt type of 7; however, the merchant in this case would be liable to a chargeback. Otherwise, the merchant can chose not to continue with the transaction. Please refer to section 9.b. for sample code and please refer to section 10 for possible liability scenarios.

7. The cardholder's browser uses the URL returned from Visa/MasterCard via the merchant to communicate directly with the bank. The contents of the inline window are loaded and the cardholder enters their PIN. Please refer to section 9.c. for sample code.

8. The information is submitted to the bank and authenticated. A response is then returned to the client browser.

9. The client browser receives the response from the bank and forwards it to the merchant.

10. The merchant receives the response information from the cardholder's browser and sends an ACS request type. For sample code please refer to section 9.d.

11. Moneris MPI receives the ACS request and authenticates the information. The Moneris MPI then provides a CAVV value to the merchant. If the "success" response is "true", the merchant may proceed with the cavv_purchase or cavv_preauth. If the response is "false" and the message is "N", the transaction must be cancelled as cardholder failed to authenticate. If the "success" response is "false" and the message is "U" or the response times out, the transaction can be processed as a normal purchase or preauth; however in this case the merchant assumes liability for a chargeback.

12. The merchant retrieves the CAVV value and formats a CAVV Purchase or CAVV PreAuth request. As part of this transaction method the merchant must pass the CAVV value. For a sample of a VbV/MCSC Purchase transaction, please refer to the complete XML DTD Integration Guide available at: http://www.eselectplus.ca/en/downloadable-content

6. Request Types

The Moneris MPI requires certain transaction requests to be submitted to complete the VbV/MCSC process. Below is a list of transactions supported by the MPI.

txn – The TXN request sends the initial transaction data to the Moneris MPI to verify if the issuer and/or card is enrolled in the Verified by Visa / MasterCard SecureCode program.

acs – The ACS request verifies whether the PIN was properly authenticated. The ACS response will return the values required to for the final step - sending a Purchase / PreAuth transaction.

7. Request DTD

<!-- The MpiRequest DTD -->

<!-- Main Elements -->

```
<!ELEMENT MpiRequest (store_id, api_token, (txn | acs)+)>
<!ELEMENT store_id (#PCDATA)>
<!ELEMENT api_token (#PCDATA)>
<!ELEMENT txn (xid, amount, pan, expdate, MD?, merchantUrl, accept, userAgent, currency?, recurFreq?, recurEnd?, install? )>
<!ELEMENT acs (PaRes, MD)>
<!ELEMENT xid (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT pan (#PCDATA)>
<!ELEMENT expdate (#PCDATA)>
<!ELEMENT MD (#PCDATA)>
<!ELEMENT merchantUrl (#PCDATA)>
<!ELEMENT accept (#PCDATA)>
<!ELEMENT userAgent (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT recurFreq (#PCDATA)>
<!ELEMENT recurEnd (#PCDATA)>
<!ELEMENT install (#PCDATA)>
<!ELEMENT PaRes (#PCDATA)>
```

8. Response DTD

<!-- The MpiResponse DTD -->

<!-- Main Elements -->

```
<!ELEMENT MpiResponse (type, success, message, PaReq, TermUrl, MD, ACSUrl, cavv, PAResVerified)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT success (#PCDATA)>
<!ELEMENT message (#PCDATA)>
<!ELEMENT PaReq (#PCDATA)>
<!ELEMENT TermUrl (#PCDATA)>
<!ELEMENT MD (#PCDATA)>
<!ELEMENT ACSUrl (#PCDATA)>
<!ELEMENT cavv (#PCDATA)>
<!ELEMENT PAResVerified (#PCDATA)>
```

9. How do I send Transactions?

A. The TXN Request (Step 2 of Transaction Flow)

The TXN request sends initial transaction data to the Moneris MPI to verify if the card / issuer is enrolled. Sample request code is below.

```
<?xml version="1.0" encoding="UTF-8"?>
<MpiRequest>
  <store_id>moneris</store_id>
  <api_token>hurgle</api_token>
  <txn>
    <xid>00000000001172089875</xid>
    <amount>4.00</amount>
    <pan>4242424242424242</pan>
    <expdate>4312</expdate>
    <MD>xid=00000000001172089875&pan=4242424242424242&expiry=4312&amount=4.00</MD>
    <merchantUrl>https://www.your_domain.com/mpistore1.cgi</merchantUrl>
    <accept>text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;
           q=0.8,image/png,*/*;q=0.5</accept>
    <userAgent>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.9)
           Gecko/20061206 Firefox/1.5.0.9</userAgent>
    <currency></currency>
    <recurFreq></recurFreq>
    <recurEnd></recurEnd>
    <install></install>
  </txn>
</MpiRequest>
```

NOTE In the example above, the merchantUrl has been highlighted. This URL is required throughout the VbV/MCSC process so that responses may be sent back to the merchant's website. Please fill in your own domain name when testing.

B. The TXN Response (Step 6 of Transaction Flow)

The TXN request will return a response with several values. The <message> will contain "Y", "U", "N". If the message is "N" the Purchase or PreAuth can be sent with a crypt_type of 6 – attempted authentication. If the message is "U" the merchant can send the transaction with crypt_type 7; however, the merchant would be liable for chargebacks. "U" is returned for non-participating cards, such as corporate cards. Finally, if the message is "Y" then you will need to create the VBV form – please refer to section 9.c. for example. Please refer to section 10 for a breakdown of possible liability scenarios.

```
<?xml version="1.0" encoding="UTF-8"?>
<MpiResponse>
  <type>VERes</type>
  <success>true</success>
  <message>Y</message>
  <PaReq>eJyNU8tuozAU3fcrENuq8SNQIDKu0smMmkXSzjTJnnGugpUAqYFJpl9fm+KYdFUkJN9zz/F9mj2c
i4P3D1QtqzL1yQj7HpSi2spyl/rr1a+72H/gN2yVK4DZK4hWAb/xPLaAus524Mlt6tMkpBG+j0ic
BPjyERJRHCdxFPpGoTUv0z/w1im60zdlWtjnx3V6I8qQNa17AUrkWdlYQEOZeHucL3lACB4nDPWm
8xeg5jM+zIChT8xxyqwA/nvq2eu9kKEOcxRRtWWj/vM4wAxZw7lbdeB50xwnCB0qkR3yqm4mMY4x
Ejt591eWaP26edyg4ijrplJARhpnyMhsbehrceylNUA9TOMst3wxm57c//O8fJ+fnmf7YPm+Thky
DMffZg1winGEKSUeCSdjPAkJQx0+aGJhKuLBCOvqesN5jyaPaU8xjCEwaFGrlN4o2yNrOQKcj1UJ
WqMnezlfyv9aLfvxdDVm0eiZqQUs8xXdrPfidp+EFc6ogFOamsF3hKtoUs8oGJPPcNINjCF7tw5r
FrRbdNRvun4E6PoVfAAyjN/F</PaReq>
  <TermUrl> https://www.your_domain.com/mpistore1.cgi</TermUrl>
  <MD>xid=00000000001172089875&pan=4242424242424242&expiry=4312&amount=4.00</MD>
  <ACSUrl>https://dropit.3dsecure.net:9443/PIT/ACS</ACSUrl>
  <cavv>null</cavv>
  <PAResVerified>null</PAResVerified>
</MpiResponse>
```

C. Creating the InLine Window

If in the TXN response (MpiResponse) the <message> contained “Y” then you will need to create the VBV form. The action URL in the form below is returned in the ACSUrl field in the MpiResponse (please see section 9.b.). The PaReq, MD and TermUrl are also returned in the MpiResponse and may be seen highlighted in the example below.

```
<html>
<head>
       <title>Title for Page</title>
</head>

<SCRIPT LANGUAGE="Javascript" >
<!--
function OnLoadEvent()
{
       document.downloadForm.submit();
}
-->
</SCRIPT>

<body onload="OnLoadEvent()">

<form name="downloadForm" action="https://dropit.3dsecure.net:9443/PIT/ACS"
       method="POST">
<noscript>
<br>
<br>
<center>
<h1>Processing your 3-D Secure Transaction</h1>
<h2>
JavaScript is currently disabled or is not supported by your browser.<br>
<h3>Please click on the Submit button to continue the processing of your 3-D secure
transaction.</h3>
<input type="submit" value="Submit">
</center>
</noscript>

<input type="hidden" name="PaReq"
       value="eJyNU8tuozAU3fcrENuq8SNQIDKu0smMmkXSzjTJnnGugpUAqYFJpl9fm+KYdFUkJN9zz/F9mj
       2ci4P3D1QtqzL1yQj7HpSi2spyl/rr1a+72H/gN2yVK4DZK4hWAb/xPLaAus524Mlt6tMkpBG+j0ic
       BPjyERJRHCdxFPpGoTUv0z/w1im60zdlWtjnx3V6I8qQNa17AUrkWdlYQEOZeHucL3lACB4nDPWm
       8xeg5jM+zIChT8xxyqwA/nvq2eu9kKEOcxRRtWWj/vM4wAxZw7lbdeB50xwnCB0qkR3yqm4mMY4x
       Ejt591eWaP26edyg4ijrplJARhpnyMhsbehrceylNUA9TOMst3wxm57c//O8fJ+fnmf7YPm+Thky
       DMffZg1winGEKSUeCSdjPAkJQx0+aGJhKuLBCOvqesN5jyaPaU8xjCEwaFGrlN4o2yNrOQKcj1UJ
       WqMnezlfyv9aLfvxdDVm0eiZqQUs8xXdrPfidp+EFc6ogFOamsF3hKtoUs8oGJPPcNINjCF7tw5r
       FrRbdNRvun4E6PoVfAAyjN/F">
<input type="hidden" name="MD"
       value="xid=00000000001172089875&pan=4242424242424242&expiry=4312&amount=4.00">
<input type="hidden" name="TermUrl"
       value=" https://www.your_domain.com/mpistore1.cgi">
</form>

</body>
</html>
```

D. The ACS Request (Step 10 of Transaction Flow)

After the cardholder completes the authentication the browser will return a response to the URL specified in the merchantUrl field which was provided in the original TXN request, please refer to section 9.a. This will contain an encoded string named PARes, as well as a success field. Once the encoded PARes is received the complete string must be passed as is to the Moneris MPI API as a type ACS.

```
<?xml version="1.0" encoding="UTF-8"?>
<MpiRequest>
   <store_id>moneris</store_id>
   <api_token>hurgle</api_token>
   <acs>
      <PaRes>eJy1V9mSqkoW/ZWK6kfjHAZR5ATljWQQURNlRt9ARmUQARm+vlNr7Op6uN0dTYRRyarcw8pcey
        fJ/tVl6dMtuFZJkb88E7/x56cgPxZ+kkcvz6ax+DV7/mvOGvE1CAQ9ODbXYM7CoKrcKHhK/JdnkpmQN
        D6liRlD4R8PQdAkPmNm9OR5zu6AFlSPycSYGRP4FGFvEeco4G+Sxd5fkevrMXbzes66x5KTlTlFEPiY
        YbG3VzYLrrIw/xqVxV4xFvs03jX3UYVy7RJ/DgXQfv7EThnkdiucKWUwX1jsPoP13TqYkzhO4yRJPBG
        TP2P8z4RgsQfOXu7uQFY0yDdiyWJfARYtyhWtWT+fUehfH29s0F2KPEAzEMGPMYt95nZx8zn+5aFICk
        29o6zhzNk6yb7kROJ/KPwPSbPYA2er2q2bag5Y7G3EHt3bbc4DwIPEJyTe19Q4UbWluQevD+L6mMIGx
        2SOT1FS6O/DCqRRcU3qOLun+q8Ai91TwR57OGf1JMpRsGvwhGSTVy/PcV1f/mBY27a/2/Hv4hphKGEc
        wxkMTfCrJPrH86tV4Mt5WPxHZrybF3lydNNkcGukDhjUceE/feT2kxtDu3siME3kfyFXv44Elf+6I=
      </PaRes>
      <MD>xid=00000000001172089875&pan=4242424242424242&expiry=4312&amount=4.00</MD>
   </acs>
</MpiRequest>
```

E. The ACS Response and forming a transaction (Step 11 of Transaction Flow)

The ACS response will contain the CAVV value. This value needs to be passed to the transaction engine along with the rest of the Purchase or PreAuth request. Please see the documentation provided by your payment solution.

```
<?xml version="1.0" encoding="UTF-8"?>
<MpiResponse>
  <type>PARes</type>
  <success>true</success>
  <message>A</message>
  <PaReq>null</PaReq>
  <TermUrl>null</TermUrl>
  <MD>null</MD>
  <ACSUrl>null</ACSUrl>
  <cavv>CAACAid1GCdRQhiQRHUYAAAAAAA=</cavv>
  <PAResVerified>true</PAResVerified>
</MpiResponse>
```

NOTE

For the XML format of a VbV/MCSC Purchase transaction ("cavv_purchase"), please refer to the complete XML DTD Integration Guide found at: http://www.eselectplus.ca/en/downloadable-content

10. What Does My VbV/MCSC Response Mean?

For each transaction, a crypt type will be sent to identify whether it is a VbV or MCSC authenticated transaction. Below are the tables defining the different possible crypt types as well as the possible VARes and PARes responses.

Crypt Type	Visa Definition	MasterCard Definition
5	- Fully authenticated - There is a liability shift and the merchant is protected from chargebacks	- Fully authenticated - There is a liability shift and the merchant is protected from chargebacks.
6	- VbV has been attempted - There is a liability shift and the Merchant is protected from chargebacks	- MCSC has been attempted - No liability shift - Merchant is not protected from chargebacks
7	- Non-VbV transaction - Merchant is no longer protected from chargebacks	- Non-MCSC transaction - No liability shift - Merchant is not protected from chargebacks

VERes Response	Response Definition
N	The card/issuer is not enrolled. This should be sent as a normal "purchase"/"preauth" transaction with a crypt type of 6.
U	The card type is not participating in VbV or MCSC: this could be corporate or other card plans that Visa or MasterCard excludes. Can proceed with a regular transaction with a crypt type of 7 or cancel the transaction.
Y	The card is enrolled. Proceed to create the VbV/MCSC inline window for cardholder authentication. Proceed to PARes for crypt type.

PARes Response	Response Definition
A	Attempted to verify PIN and will receive a CAVV. This should now be sent as a "cavv_purchase"/"cavv_preauth" which will return a crypt type of 6.
Y	Fully authenticated and will receive a CAVV. This should now be sent as a "cavv_purchase"/"cavv_preauth" which will return a crypt type of 5.
N	Failed to authenticate, no CAVV will be returned. Transaction should be cancelled; the merchant may proceed with a crypt type of 7 though strongly discouraged.

Step 1: VERes Is the cardholder/issuer enrolled?	Step 2: PARes VbV/MCSC InLine window response	Step 3: Transaction Are you protected?
Y	Y	Send a CAVV transaction
Y	N	Send a regular transaction with a crypt type of 7 (strongly encouraged to cancel)
Y	A	Send a CAVV transaction
U	n/a	Send a regular transaction with a crypt type of 7
N	n/a	Send a regular transaction with a crypt type of 6

11. How Do I Test My Solution?

When testing your implementation of the Moneris MPI you can use the VISA / MASTERCARD PIT (production integration testing) environment to test. When testing, the process is a little different in that when the inLine window is generated it will not contain any input boxes but rather a window of data and a "Submit" button. When you hit "Submit" it will load the response in the window and not in the main as it will in production.

The test environment is generally available 7x24, however since it is a test environment we cannot guarantee 100% availability. Also, please be aware that other merchants are using the test environment so in the Merchant Resource Centre you may see transactions and user IDs that you did not create. As a courtesy to others that are testing we ask that when you are processing Refunds, changing passwords and/or trying other functions that you use only the transactions/users that you created.

When using the Moneris MPI in the test environment you will need to use the test store_id and api_token. These are different than your production IDs. The IDs that you can use in the test environment are in the table below.

Test IDs			
store_id	**api_token**	**Username**	**Password**
moneris	hurgle	demouser	password

When testing you may use the following test card numbers with any future expiry date.

Test Card Numbers			
VERes	**PARes**	**Card Number**	**Action**
Y	true	4012001037141112	TXN – call function to create inLine window ACS – Send CAVV to eSELECTplus using "cavv_purchase" or "cavv_preauth"
U	N/A	4012001038488884	Send transaction to eSELECTplus using regular "purchase" or "preauth". Set crypt_type = 7
N	N/A	4012001038443335	Send transaction to eSELECTplus using regular "purchase" or "preauth". Set crypt_type = 6
Y	true	4242424242424242	TXN – call function to create inLine window ACS – Send CAVV to eSELECTplus using "cavv_purchase" or "cavv_preauth"
Y	false	4012001037461114	Card failed to authenticate, merchant may chose to send transaction or decline transaction. If transaction is sent crypt type = 7

VERes – the result U, Y or N is obtained by parsing the data within the `<message>` tags
PARes – the result "true" or "false" is obtained by parsing the data within the `<success>` tags

---

NOTE

For the XML format of a "purchase", "preauth", "cavv_purchase" and "cavv_preauth" transaction, please refer to the complete XML DTD Integration Guide found at: http://www.eselectplus.ca/en/downloadable-content

---

To access the Merchant Resource Centre in the test environment go to https://esqa.moneris.com/mpg . And use the logins provided in the previous table.

The test environment has been designed to replicate our production environment as closely as possible. One major difference is that we are unable to send test transactions onto the production authorization network and thus Issuer responses are simulated. Additionally, the requirement to emulate approval, decline and error situations dictates that we use certain transaction variables to initiate various response and error situations.

The test environment will approve and decline transactions based on the penny value of the amount field.
For example, a transaction made for the amount of $9.00 or $1.00 will approve since the .00 penny value is set to approve in the test environment. Transactions in the test environment should not exceed $1000.00. This limit does not exist in the production environment. For a list of all current test environment responses for various penny values, please see the Test Environment Penny Response table as well as the Test Environment eFraud Response table, available at http://www.eselectplus.ca/en/downloadable-content

NOTE These responses may change without notice. Moneris Solutions recommends you regularly refer to our website to check for possible changes.

12. How Do I Configure My Store For Production?

Once you have completed testing using the PIT you are ready to point your store to the production host. You will need to change the post URL as highlighted below in red. You will also need to change the store_id to reflect your production store ID as well the api_token must be changed to your production token to reflect the token that you received during activation.

Production	`https://www3.moneris.com/gateway2/servlet/MpgRequest`
Development	`https://esqa.moneris.com/gateway2/servlet/MpgRequest`

13. How Do I Get Help?

If you require technical assistance while integrating your store, please contact the eSELECTplus Support Team:

For technical support:
Phone: 1-866-319-7450 (Technical Difficulties)

For integration support:
Phone: 1-866-562-4354
Email: eselectplus@moneris.com

When sending an email support request please be sure to include your name and phone number, a clear description of the problem as well as the type of API that you are using. **For security reasons, please do not send us your API Token combined with your store ID, or your merchant number and device number in the same email.**

14. Appendix A. Definition of Required Fields

Required Fields		
Variable Name	**Size/Type**	**Description**
store_id	12 / an	A value that identifies your company when you send a transaction. Provided by Moneris Solutions.
api_token	20 / an	A unique key that when matched with your store_id creates a secure method of authenticating your store_id. Generated when you first activate your store.
xid	20 / an	Must be exactly 20 alpha numeric characters. This must be unique for every transaction attempt – this can also be used as your order_id when using the eSELECTplus MPG XML DTD
amount	9 / decimal	Amount of the transaction. This must contain at least 3 digits including two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
pan	20 / num	Credit Card Number - no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration for future expansion and/or potential support of private label card ranges.
expdate	4 / num	Expiry Date - format **YYMM** no spaces or slashes. PLEASE NOTE THAT THIS IS REVERSED FROM THE DATE DISPLAYED ON THE PHYSICAL CARD WHICH IS MMYY
MD	1024 / an	This is information that you would like echoed back in the response
merchantUrl		This is the URL to which you would like the MPI response sent to. Please refer to section 9.a. and 9.d.
accept		MIME types the browser accepts
userAgent		The browser details
PaRes		This is a value that is passed back to the API during the TXN and returned to the MPI when an ACS request is made. Please refer to section 9.d. for further details.

15. Appendix B. Definition of Response Fields

<table>
<tr><th colspan="3">Response Fields</th></tr>
<tr><th>Variable Name</th><th>Size/Type</th><th>Description</th></tr>
<tr><td>type</td><td>99 / an</td><td>VERes or PARes or error defines what type of response you are receiving</td></tr>
<tr><td>success</td><td>true/false</td><td>Returns whether the attempt was successful or not</td></tr>
<tr><td>message</td><td>alpha</td><td>Will contain:
<table>
<tr><th>Txn</th><th>Action</th></tr>
<tr><td>Y</td><td>Create VBV verification form popup window.</td></tr>
<tr><td>N</td><td>Send purchase or preauth with crypt type 6</td></tr>
<tr><td>U</td><td>Send purchase or preauth as crypt type 7</td></tr>
</table>
<table>
<tr><th>ACS</th><th>Action</th></tr>
<tr><td>Y or A
(<success>true</success>)</td><td>Proceed with cavv purchase or cavv preauth</td></tr>
<tr><td>N</td><td>Transaction must be cancelled, authentication failed.</td></tr>
<tr><td>U or time out</td><td>Send purchase or preauth as crypt type 7</td></tr>
</table>
</td></tr>
<tr><td>PARes</td><td>n/a</td><td>Variable Length – data that Visa/MasterCard passes and needs for authentication</td></tr>
<tr><td>TermUrl</td><td>255 / an</td><td>The URL to which the PARes is returned. Please refer to section 9.d.</td></tr>
<tr><td>MD</td><td>1024 / an</td><td>Merchant defined data that will be echoed back</td></tr>
<tr><td>ACSUrl</td><td>255 / an</td><td>URL that will generate the inLine window</td></tr>
<tr><td>cavv</td><td>28 /an</td><td>Visa/MasterCard authentication field</td></tr>
</table>