Merchant Integration Guide

COM API – Vault

V 1.1.6

Moneris SOLUTIONS

Revision Number	Date	Changes
V1.1.1	May 5, 2009	-Document edited for coherence
V1.1.2	June 20, 2011	-New download link updated in various locations https://esplusqa.moneris.com/connect/en/download/index.php -Section 2. System and Skill Requirements -Added PCI & PA DSS note.
V1.1.3	December 13, 2011	-Appendix I. Card Validation Digits (CVD) -Added American Express/JCB response codes -Appendix J. Address Verification Service (AVS) -Added American Express/JCB response codes -Appendix K. Additional Information for CVD and AVS -Added American Express/JCB response codes
V1.1.4	March 29, 2012	-Appendix A. Definition of Request Fields -Added new Variable Name (dynamic_descriptor) -Appendix D. Recur Fields - Added recur_unit End of Month. - Corrected recur examples
V1.1.5	September 6, 2012	-New download link updated in various locations: https://developer.moneris.com/ -Section 4. Transaction Types – Added ResTokenizeCC -Section 5. Administrative Transactions – Added ResTokenizeCC example -Appendix F. ProgIDs – Added Moneris.USResTokenizeCC -Appendix G. Method List – Added Moneris.USResTokenizeCC
V1.1.6	November 13, 2012	-Section 4. Transaction Types – Added EncResAddCC & EncResUpdateCC -Section 5. Administrative Transactions - Added EncResAddCC example - Added EncResUpdateCC example -Section 7. Financial Transaction with Extra features: ResPurchaseCC with CVD and AVS (eFraud) -Added CVD note. -Appendix A. Definition of Request Fields – Added new variable names (enc_track2 & device_type) – Added CVD note. -Appendix F. ProgIDs – Added Moneris.USEncResAddCC & Moneris.USEncResUpdateCC -Appendix G. Method List – Added Moneris.USEncResAddCC & Moneris.USEncResUpdateCC -Appendix I. Card Validation Digits (CVD) – Added CVD note.

Table of Contents

**** PLEASE READ CAREFULLY****

You have a responsibility to protect cardholder and merchant related confidential account information. Under no circumstances should ANY confidential information be sent via email while attempting to diagnose integration or production issues. When sending sample files or code for analysis by Moneris staff, all references to valid card numbers, merchant accounts and transaction tokens should be removed and or obscured. Under no circumstances should live cardholder accounts be used in the test environment.

1. About this Documentation

This document describes the basic information for using the COM API for sending Vault transactions as well as outlining all administrative functions of the Vault. The Vault feature allows a merchant to create customer profiles, edit those profiles, and use them to process transactions without having to enter financial information each time. This document will outline all the steps required in order to fully utilize this functionality and will not describe basic transaction processing. To access basic transaction processing information without the Vault, for example Refund and Void, please refer to the COM API Integration Guide available at: https://developer.moneris.com

2. System and Skill Requirements

In order to use COM your system will need to have the following:
1. Port 443 open
2. Web server with an SSL certificate
3. WinHTTP 5.1 (Please note that this will not work with winHTTP 5.0)
(Note: Core SDK is no longer required)

As well, you will need to have the following knowledge and/or skill set:
1. Permissions to register a dll
2. Knowledge of a COM compatible language (ASP or VBscipt)

Note:

It is important to note that all Merchants and Service Providers that store, process, or transmit cardholder data must comply with PCI DSS and the Card Association Compliance Programs. However, certification requirements vary by business and are contingent upon your "Merchant Level" or "Service Provider Level". Failure to comply with PCI DSS and the Card Association Compliance Programs may result in a Merchant being subject to fines, fees or assessments and/or termination of processing services. Non-compliant solutions may prevent merchants boarding with Moneris Solutions.

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.

For further information on PCI DSS and PA DSS requirements, please visit http://www.pcisecuritystandards.org.

For more information on how to get your application PCI-DSS compliant, please contact our Integration Specialists and visit https://developer.moneris.com to download the PCI-DSS Implementation Guide.

3. 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. Refer to the main COM API Integration Guide to develop all follow-on procedures (ex. Refund) https://developer.moneris.com .
3. Test your solution in the test environment.
4. Activate your store.
5. Make the necessary changes to move your solution from the test environment into production as outlined in this document.

4. Transaction Types

The Vault API supports both financial and administrative transactions. These transactions are outlined below.

Vault Transactions (Admin)

ResAddCC – Create a new credit card profile. The fields which may be sent in are outlined in the transaction examples which can be found in section 5 of this documentation.

EncResAddCC – Similar to the regular ResAddCC, the Encrypted ResAddCC transaction type creates a new credit card profile. This transaction type requires the card data to be either swiped or manually keyed in via a Moneris provided encrypted mag swipe reader.

ResTokenizeCC - Create a new credit card profile using the credit card number and expiry date submitted in a previous financial transaction. The fields which may be sent in are outlined in the transaction examples which can be found in section 5 of this documentation.

ResAddAch – Create a new ACH profile. The fields which may be sent in are outlined in the transaction examples which can be found in section 5 of this documentation.

ResAddPinless – Create a new Pinless Debit profile. The fields which may be sent in are outlined in the transaction examples which can be found in section 5 of this documentation.

ResUpdateCC – This will update a profile to contain Credit Card information using a unique data_key. If the profile which is being updated was already a Credit Card profile, all information contained within it will simply be updated as indicated by the submitted fields. If however the profile was of a different payment type (ie: ACH or Pinless Debit), the old profile will be deactivated and the new Credit Card information will be associated with the data_key. As a result, the mandatory fields for creating a new Credit Card profile will be required. These are all outlined in the transaction examples found in section 5 of this documentation.

EncResUpdateCC – Similar to the regular ResUpdateCC, the Encrypted ResUpdateCC transaction type will update a profile to contain Credit Card information using a unique data_key. This transaction type requires the card data to be either swiped or manually keyed in via a Moneris provided encrypted mag swipe reader.

ResUpdateAch – This will update a profile to contain ACH information using a unique data_key. If the profile which is being updated was already an ACH profile, all information contained within it will simply be updated as indicated by the submitted fields. If however the profile was of a different payment type (ie: Credit Card or Pinless Debit), the old profile will be deactivated and the new ACH information will be associated with the data_key. As a result, the mandatory fields for creating a new ACH profile will be required. These are all outlined in the transaction examples found in section 5 of this documentation.

ResUpdatePinless – This will update a profile to contain Pinless Debit Card information using a unique data_key. If the profile which is being updated was already a Pinless Debit Card profile, all information contained within it will simply be updated as indicated by the submitted fields. If however the profile was of a different payment type (ie: Credit Card or ACH), the old profile will be deactivated and the new Pinless Debit Card information will be associated with the data_key. As a result, the mandatory fields for creating a new Pinless Debit Card profile will be required. These are all outlined in the transaction examples found in section 5 of this documentation.

ResDelete – Delete an existing profile of any payment type using the unique data_key which was assigned when the profile was first added. *It is important to note that once a profile is deleted, the information which was saved within can no longer be retrieved.*

ResGetExpiring – Retrieve all Credit and Pinless Debit cards which are about to expire, as well as the Vault data which is associated with each profile. This transaction will retrieve cards which will expire within the current calendar month or one month following. This transaction will be limited to being performed a maximum of 2 times per calendar day.

ResLookupMasked – Retrieve all Vault data that is associated with a unique data_key. The Credit Card, Pinless Debit Card or bank account number that will be returned will be masked.

ResLookupFull – Retrieve all Vault data that is associated with a unique data_key. Unlike ResLookupMasked, this transaction will return both the full unmasked Credit Card, Pinless Debit Card or bank account number as well as the masked value.

Vault Transactions (Financial)

ResPreauthCC – This is a preauthorization transaction for Credit Card profiles only. This transaction will use a unique data_key which will identify a previously registered Credit Card profile. The details within the profile will be submitted to perform the preauthorization transaction.

ResPurchase(CC|ACH|Pinless) – This is a purchase transaction which can be used for all the payment types. For Credit Cards, this is processed as a us_purchase transaction. For ACH this is processed as a us_ach_debit. For Pinless Debit, this is processed as a us_pinless_debit_purchase transaction. The name of the transaction (ex. ResPurchaseCC) must coincide with the payment type associated with the data_key (ex.Credit Card).

ResIndRefund(CC|ACH) – This is an independent refund transaction which can be used for Credit Card and ACH profiles only. For ACH transactions, this is processed as a us_ach_credit. The name of the transaction (ex. ResIndRefundCC) must coincide with the payment type associated with the data_key (ex.Credit Card).

5. Administrative Transactions

Included below is the sample code for the Administrative transactions that can be found in the "Examples" folder of the Vault COM API download. Administrative transactions allow the user to perform such tasks as creating new Vault profiles, deleting existing profiles and updating profile information and payment types.

ResAddCC

This transaction is used to create a new Credit Card profile. A unique data_key will be generated and returned to the merchant in the response. This will be the identifier for this profile which all other Vault financial transactions will use in order to associate the transaction with the saved information. Please refer to sections 4 and 6 for examples of the financial transactions available.

The mandatory fields for this transaction are: pan, expdate, crypt_type (required to register a CC transaction but will not be used for any Vault financial transactions). Optional fields are: avs_info, cust_id, email, phone, and note. The ResolveData that is returned in the response will indicate the fields registered for this profile.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        cust_id = "602" & Time()
        phone = "999-888-7777"
        email = "my@email.com"
        note = "hello world"
        pan = "5454545442424242"
        expdate = "1212"
        crypt_type = "7"

        avs_street_number = "6600"
        avs_street_name = "New York Street"
        avs_zipcode = "90210"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set resaddccrequest = server.CreateObject("Moneris.USResAddCC")
            resaddccrequest.setAvsInfo avs_street_number, avs_street_name, avs_zipcode

        out.setRequest resaddccrequest.formatRequest(cust_id, phone, email, note, pan, expdate, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
```

```
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

EncResAddCC

Similar to the standard ResAddCC transaction type, the EncResAddCC transaction is used to create a new Credit Card profile. The EncResAddCC transaction allows the merchant to swipe or manually key in the credit card details using a Moneris provided encrypted reader and submit the encrypted track2 details. A unique data_key will be generated and returned to the merchant in the response. This will be the identifier for this profile which all other Vault financial transactions will use in order to associate the transaction with the saved information. Please refer to sections 4 and 6 for examples of the financial transactions available.

The mandatory fields for this transaction are: enc_track2, device_type, and crypt_type (required to register a CC transaction but will not be used for any Vault financial transactions). Optional fields are: AvsInfo, cust_id, email, phone, and note. The ResolveData that is returned in the response will indicate the fields registered for this profile.

NOTE Please note, the Encrypted Transactions may only be used with a Moneris provided encrypted mag swipe reader. To enquire about the encrypted MSR, please call the Service Centre at 1-866-423-8475.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        cust_id = "602" & Time()
        phone = "999-888-7777"
        email = "my@email.com"
        note = "hello world"
        enc_track2 =
                "02D901801F4F2800039B%*4924********4030^TESTCARD/MONERIS^****************************************?*;492
                4********4030=********************?*A7150C78335A5024949516FDA9A68A91C4FBAB1279DD1DE2283DBEBB2C6B3FDEACF7
                B5B314219D76C00890F347A9640EFE90023E31622F5FD95C14C0362DD2EAB28ADEB46B8B577DA1A18B707BCC7E48068EFF1882CF
                B4B369BDC4BB646C870D6083239860B23837EA91DB3F1D8AD066DAAACE2B2DA18D563E4F1EF997696337B8999E9C707DEC4CB041
                0B887291CAF2EE449573D01613484B80760742A3506C31415939320000A000283C5E03"
        device_type = "idtech"
        crypt_type = "1"

        avs_street_number = "6600"
        avs_street_name = "New York Street"
        avs_zipcode = "90210"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set encresaddccrequest = server.CreateObject("Moneris.USEncResAddCC")
        encresaddccrequest.setAvsInfo avs_street_number, avs_street_name, avs_zipcode
        out.setRequest encresaddccrequest.formatRequest(cust_id, phone, email, note, enc_track2, device_type, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "dumpXML:  " & out.dumpXMLResponse & "<br>"
%>
</BODY>
</HTML>
```

ResTokenizeCC

The ResTokenizeCC transaction is used to create a new Credit Card profile, but using the credit card number, expiry date and crypt type from a previous financial transaction. Similarly to a ResAddCC, a unique data_key will be generated and returned to the merchant in the response. This will be the identifier for this profile which all other Vault financial transactions will use in order to associate the transaction with the saved information.

The mandatory fields for this transaction are: order_id, txn_number (These fields are required to reference a previously processed credit card financial transaction. The credit card number, expiry date, and crypt type from this transaction will be registered in the Vault for future Vault financial transactions). Optional fields are: avs_info, cust_id, email, phone, and note. The ResolveData that is returned in the response will indicate the fields registered for this profile.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "602" & Time()
        txn_number = "123456-0_1"
        phone = "1-777-888-9999"
        email = "i.have.no@email.com"
        note = "how are you"
        cust_id="cus_id"

        street_num = "123"
        street_name = "Main Street"
        zipcode = "90210"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResTokenizeCC")
        request1.setCustId cust_id
        request1.setPhone phone
        request1.setNote note
        request1.setEmail email
        request1.setAvsInfo street_num, street_name, zipcode

        out.setRequest request1.formatRequest(order_id, txn_number)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"

        Response.Write "dumpXML:  " & out.dumpXMLResponse & "<br>"
%>

</BODY>
</HTML>
```

ResAddAch

This transaction is used to create a new ACH profile. A data_key will be generated and returned to the merchant in the response. This will be the identifier for this profile which all other Vault financial transactions will use in order to associate the transaction with the saved information. Please note, only the following SEC codes are currently supported: PPD, CCD, and WEB. The SEC code, as well as the rest of the ach_info data, that is registered will be submitted with all future Vault transactions unless it is later updated. Mandatory fields are: sec, routing_num, account_num, account_type. Optional fields are: phone, email, note, cust_id and all other ach_info fields.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        cust_id = "602" & Time()
        phone = "999-888-7777"
        email = "my@email.com"
        note = "hello world"
        sec = "ccd"
        cust_first_name = "Bob"
        cust_last_name = "Smith"
        cust_address1 = "101 Main St"
        cust_address2 = "Apt 102"
        cust_city = "Chicago"
        cust_state = "IL"
        cust_zip = "123456"
        routing_num = "543211234"
        account_num = "23456"
        check_num = "100"
        account_type = "checking"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set resaddachrequest = server.CreateObject("Moneris.USResAddAch")
        resaddachrequest.setAchInfo sec, cust_first_name, cust_last_name, cust_address1, cust_address2,
                        cust_city, cust_state, cust_zip, routing_num, account_num, check_num, account_type
        out.setRequest resaddachrequest.formatRequest(cust_id, phone, email, note)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResAddPinless

This transaction is used to create a new Pinless Debit profile. A data_key will be generated and returned to the merchant in the response. This will be the unique identifier for this profile which all other Vault financial transactions will use in order to associate the transaction with the saved information. The presentation_type that is registered will be submitted with all future Vault financial transactions, unless it is later updated. Mandatory fields are pan and presentation_type. Optional fields are email, note, phone, cust_id, expdate, and p_account_number.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        cust_id = "602" & Time()
        phone = "999-888-7777"
        email = "my@email.com"
        note = "hello world"
        pan = "4496270000164824"
        presentation_type = "W"

        expdate = "1212"
        p_account_number = "123456789012345678901234 5"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResAddPinless")

        out.setRequest request1.formatRequest(cust_id, phone, email, note, pan, presentation_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
%>

</BODY>
</HTML>
```

ResDelete

This transaction is used to delete an existing Vault profile. The data_key from the original profile will be required for this transaction. Within the ResolveData of the response, all details that were associated with the profile will be returned. Please note, the full card number or account number will not be returned. Please refer to the ResLookupFull transaction to see how to retrieve these details prior to deleting the profile.

Please note: Once a profile is deleted, the details can no longer be retrieved

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
	store_id = "monusqa002"
	api_token = "qatoken"
	data_key = "S5Yf9BTPvyoKMmuXn8YFhmM7R"

	Set out = server.CreateObject("Moneris.USResolverRequest")
	out.initRequest store_id, api_token, "esplusqa.moneris.com"

	Set request1 = server.CreateObject("Moneris.USResDelete")

	out.setRequest request1.formatRequest(data_key)
	out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
	Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
	Response.Write "Response Code:  " & out.getResponseCode & "<br>"
	Response.Write "Transaction Type:  " & out.getTransType & "<br>"
	Response.Write "Message:  " & out.getMessage & "<br>"
	Response.Write "Amount:  " & out.getTransAmount & "<br>"
	Response.Write "TransDate:  " & out.getTransDate & "<br>"
	Response.Write "TransTime:  " & out.getTransTime & "<br>"
	Response.Write "Complete:  " & out.getComplete & "<br>"
	Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

	Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
	Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
	Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
	Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
	Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
	Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
	Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
	Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
	Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
	Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
	Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
	Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
	Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
	Response.Write "Sec:  " & out.getResDataSec & "<br>"
	Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
	Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
	Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
	Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
	Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
	Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
	Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
	Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
	Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
	Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
	Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResUpdateCC

This transaction is used to update an existing Vault profile. The ResUpdateCC transaction pertains to Credit Cards and will update the existing profile accordingly. If the profile is of a different payment type, it will be automatically deactivated and a new Credit Card profile will be created and assigned to the data_key. The only data that will remain from the prior profile is the cust_id, phone, email and note associated with this data_key. For example, if data_key 'abc' refers to an ACH profile but it is submitted in the ResUpdateCC, the ach_info details will be deactivated and the new CC details will be registered. In this example, because the payment type is being changed, the following fields would be mandatory: pan, expdate, crypt_type. Otherwise, if the payment type is not being changed, all fields are optional besides the data_key. If a field is submitted, it will be updated. For example, if a blank field is submitted in cust_id, the cust_id will be deleted. The ResolveData will return all the details that are associated with the profile *after* the update.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "602" & Time()
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        phone = "1-777-888-9999"
        email = "i.have.no@email.com"
        note = "how are you"
        pan = "4242424254545454"
        exp_date = "1010"
        crypt_type = "1"
        cust_id="cus_id"

        avs_street_number = ""
        avs_street_name = ""
        avs_zipcode = "98765"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResUpdateCC")
            request1.setAvsInfo avs_street_number, avs_street_name, avs_zipcode
            request1.setCustId cust_id
            request1.setPhone phone
            request1.setEmail email
            request1.setNote note
            request1.setPan pan
            request1.setExpdate exp_date
            request1.setCryptType crypt_type
        out.setRequest request1.formatRequest(data_key)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
%>
</BODY>
</HTML>
```

EncResUpdateCC

Similar to the standard ResUpdateCC transaction, the EncResUpdateCC is used to update an existing Vault profile. The EncResUpdateCC transaction pertains to Credit Cards and will update the existing profile accordingly. The EncResAddCC transaction allows the merchant to swipe or manually key in the credit card details using a Moneris provided encrypted reader and submit the encrypted track2 details.

If the profile is of a different payment type, it will be automatically deactivated and a new Credit Card profile will be created and assigned to the data_key. The only data that will remain from the prior profile is the cust_id, phone, email and note associated with this data_key. For example, if data_key 'abc' refers to an ACH profile but it is submitted in the EncResUpdateCC, the ACHInfo details will be deactivated and the new CC details will be registered. In this example, because the payment type is being changed, the following fields would be mandatory: enc_track2, device_type, crypt_type and data_key. Otherwise, if the payment type is not being changed, only the enc_track2, device_type and data_key are mandatory and all other fields are optional. If a field is submitted, it will be updated. For example, if a blank field is submitted in cust_id, the cust_id will be deleted. The ResolveData will return all the details that are associated with the profile *after* the update.

NOTE Please note, the Encrypted Transactions may only be used with a Moneris provided encrypted mag swipe reader. To enquire about the encrypted MSR, please call the Service Centre at 1-866-423-8475.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "602" & Time()
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        phone = "1-777-888-9999"
        email = "i.have.no@email.com"
        note = "how are you"
        enc_track2 =
                "02D901801F4F2800039B%*4924********4030^TESTCARD/MONERIS^****************************************?*;492
                4********4030=********************?*A7150C78335A5024949516FDA9A68A91C4FBAB1279DD1DE2283DBEBB2C6B3FDEACF7
                B5B314219D76C00890F347A9640EFE90023E31622F5FD95C14C0362DD2EAB28ADEB46B8B577DA1A18B707BCC7E48068EFF1882CF
                B4B369BDC4BB646C870D6083239860B23837EA91DB3F1D8AD066DAAACE2B2DA18D563E4F1EF997696337B8999E9C707DEC4CB041
                0B887291CAF2EE449573D01613484B80760742A3506C31415939320000A000283C5E03"
        device_type = "idtech"
        exp_date = "1010"
        crypt_type = "1"
        cust_id="cus_id"

        avs_street_number = ""
        avs_street_name = ""
        avs_zipcode = "98765"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USEncResUpdateCC")
                request1.setAvsInfo avs_street_number, avs_street_name, avs_zipcode
                request1.setCustId cust_id
                request1.setPhone phone
                request1.setEmail email
                request1.setNote note
                request1.setCryptType crypt_type
        out.setRequest request1.formatRequest(data_key, enc_track2, device_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
```

```
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "dumpXML:  " & out.dumpXMLResponse & "<br>"
%>

</BODY>
</HTML>
```

ResUpdateAch

This transaction is used to update an existing Vault profile. The ResUpdateAch transaction pertains to ACH details and will update the existing profile accordingly. If the profile is of a different payment type, it will be automatically deactivated and a new ACH profile will be created and assigned to the data_key. A full explanation of how this update will behave can be found in the ResUpdateCC example above.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "602" & Time()
        data_key = "LlPO7PJqBvxeuqjQFKhAbVkYl"
        phone = "1-777-888-9999"
        email = "i.have.no@email.com"
        note = "how are you"
        sec = "web"
        routing_num = "54321"
        account_num = "23456"
        account_type = "checking"
        check_num = "100"
        cust_first_name = "Jerry"
        cust_last_name = "Smythe"
        cust_address1 = ""
        cust_address2 = ""
        cust_city = ""
        cust_state = ""
        cust_zip = ""
        cust_id="cus_id"
        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set request1 = server.CreateObject("Moneris.USResUpdateAch")
            request1.setAchInfo sec, cust_first_name, cust_last_name, cust_address1, cust_address2,
                        cust_city, cust_state, cust_zip, routing_num, account_num, check_num, account_type
            request1.setCustId cust_id
            request1.setPhone phone
            request1.setEmail email
            request1.setNote note
        out.setRequest request1.formatRequest(data_key)
        out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResUpdatePinless

This transaction is used to update an existing Vault profile. The ResUpdatePinless transaction pertains to Pinless Debit and will update the existing profile accordingly. If the profile is of a different payment type, it will be automatically deactivated and a new Pinless Debit profile will be created and assigned to the data_key. A full explanation of how this update will behave can be found in the ResUpdateCC example above.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "602" & Time()
        data_key = "sbZvOEknuYjd6sarHyX4kdcal"
        phone = "1-777-888-9999"
        email = "i.have.no@email.com"
        note = "how are you"
        pan = "4496270000164824"
        presentation_type = "W"
        exp_date = ""
        p_account_number = ""
        cust_id="cust_id"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResUpdatePinless")
            request1.setCustId cust_id
            request1.setPhone phone
            request1.setEmail email
            request1.setNote note
            request1.setPan pan
            request1.setExpdate exp_date
            request1.setPresentationType presentation_type
            request1.setPAccountNumber p_account_number
        out.setRequest request1.formatRequest(data_key)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResLookupFull

This transaction is used to verify what is currently saved under a given Vault profile. The data_key for the profile will need to be provided for this transaction. The response will return the latest active data for the given data_key. The ResLookupFull transaction returns the full pan or account_num as well as the masked_pan or masked_account_num.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "Lookup-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResLookupFull")
        out.setRequest request1.formatRequest(data_key)
        out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataPan & "<br>"

        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResLookupMasked

This transaction is used to verify what is currently saved under a given Vault profile. The data_key for the profile will need to be provided for this transaction. The response will return the latest active data for the given data_key. The ResLookupMasked transaction returns the card number however only first 4 and last 4 digits, or the masked account number depending on the payment type. Please refer to the ResLookupFull transaction to retrieve the un-masked details from the profile.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "602" & Time()
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResLookupMasked")
        out.setRequest request1.formatRequest(data_key)
        out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResGetExpiring

This transaction is used to verify which profiles will be expiring within the current month and one month following. For example, if processing this transaction on September 2nd 2008, then it will return all cards expiring in September and October of 2008. This particular transaction can only be performed a maximum of 2 times in any given calendar day, and it only applies to Credit Card and Pinless Debit profiles. Please note, any Pinless Debit profile which does not have an expiry date registered will not be returned in the ResGetExpiring transaction. The response will provide all expiring cards as well as the details registered in their profile for the specified store_id.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
	store_id = "monusqa002"
	api_token = "qatoken"

	Set out = server.CreateObject("Moneris.USResolverRequest")
	out.initRequest store_id, api_token, "esplusqa.moneris.com"

	Set request1 = server.CreateObject("Moneris.USResGetExpiring")

	out.setRequest request1.formatRequest()
	out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
	Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
	Response.Write "Response Code:  " & out.getResponseCode & "<br>"
	Response.Write "Transaction Type:  " & out.getTransType & "<br>"
	Response.Write "Message:  " & out.getMessage & "<br>"
	Response.Write "Amount:  " & out.getTransAmount & "<br>"
	Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
	Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
	Response.Write "ResDataDataKey:  " & out.getResDataDataKey & "<br>"
	Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
	Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
	Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
	Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
	Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
	Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
	Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
	Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
%>

</BODY>
</HTML>
```

6. Financial Transaction Examples

Included below is the sample code that can be found in the "Examples" folder of the Vault COM API download. Vault transactions are very similar to regular financial transactions. The main difference is the use of a data_key which is used as a reference to all the mandatory financial information normally found in a regular transaction. It is important to note that the transaction type used must match the payment type which is saved in the profile. For example, a ResPurchaseCC transaction may not use a data_key which references an ACH profile. Once the transaction is complete, the response will also include all the fields which are currently saved under the profile which was used. It is also important to note that cust_id is not a mandatory variable. If it is passed in, it will be used for the current transaction. If cust_id is not passed in and there is a cust_id saved in the customer profile, the profile cust_id will then be used. If no cust_id is passed in and there is none saved in the customer profile, the transaction will be completed without a cust_id.

ResPreauthCC (basic)

The ResPreauthCC transaction will process a us_preauth transaction for a Credit Card using saved information in a Vault profile. In the ResPreauthCC example we require several variables (store_id, api_token, data_key, order_id, amount, and crypt_type). If avs_info is registered in the profile, it will be submitted with the us_preauth as well as returned in the ResolveData portion of the response. EFraud is outlined in greater detail in the following section. Please refer to Appendix A. Definition of Request Fields for variable definitions.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
	store_id = "monusqa002"
	api_token = "qatoken"

	order_id = "resPreauth-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
	data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
	amount = "1.00"
	crypt_type = "1"

	Set out = server.CreateObject("Moneris.USResolverRequest")
	out.initRequest store_id, api_token, "esplusqa.moneris.com"

	Set request1 = server.CreateObject("Moneris.USResPreauthCC")

	out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
	out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
	Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
	Response.Write "Response Code:  " & out.getResponseCode & "<br>"
	Response.Write "Transaction Type:  " & out.getTransType & "<br>"
	Response.Write "Message:  " & out.getMessage & "<br>"
	Response.Write "Amount:  " & out.getTransAmount & "<br>"
	Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
	Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
	Response.Write "Transaction ID:  " & out.getTransID & "<br>"
	Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
	Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
	Response.Write "Card Type:  " & out.getCardType & "<br>"
	Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
	Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"

	Response.Write "TransDate:  " & out.getTransDate & "<br>"
	Response.Write "TransTime:  " & out.getTransTime & "<br>"
	Response.Write "Complete:  " & out.getComplete & "<br>"
	Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
```

```
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

ResPurchaseCC (basic)

The ResPurchaseCC transaction will process a us_purchase transaction for a Credit Card using saved information in a Vault profile. In the ResPurchaseCC example we require several variables (store_id, api_token, data_key, order_id, amount, and crypt_type). Optional variables are cust_id, commcard_invoice and commcard_tax_amount. The commcard_* variables should be passed in blank if not used.). If avs_info is registered in the profile, it will be submitted with the us_purchase as well as returned in the ResolveData portion of the response. EFraud is outlined in greater detail in the following section. Please refer to Appendix A. Definition of Request Fields for variable definitions.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "602" & Time()
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        amount = "1.00"
        crypt_type = "1"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set request1 = server.CreateObject("Moneris.USResPurchaseCC")
        out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResPurchaseAch (basic)

In the ResPurchaseAch example we require several variables (store_id, api_token, data_key, order_id, amount). The optional variable is cust_id. This transaction will be processed as an us_ach_debit. The ach_info registered for this profile will be used. The details submitted within 'ach_info' will be returned in the response within ResolveData. Please refer to Appendix A. Definition of Request Fields for variable definitions.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "602" & Time()
        data_key = "Mu0SL5tTX1t9VrCfl5G4PosZC"
        amount = "1.00"
        cust_id = "customer 2"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPurchaseAch")
            request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataCustLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResPurchasePinless (basic)

In the ResPurchasePinless example we require several variables (store_id, api_token, data_key, order_id, amount, intended_use). Optional variables are cust_id and p_account_number. If p_account_number is sent, it will be submitted with the purchase but not stored in the profile. If however it is not sent, the p_account_number will be pulled from the profile. If no p_account_number is sent or found in the profile, an error will be returned. Please refer to Appendix A. Definition of Request Fields for variable definitions.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "resPurch-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "RJGjZ6ajb4mT9VNM6whYLZX4l"
        amount = "1.00"
        intended_use = "1"
        p_account_number = "234567899876543212345678 9"
        cust_id = "customer 2"
        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set request1 = server.CreateObject("Moneris.USResPurchasePinless")
                request1.setPAccountNumber p_account_number
                request1.setCustId cust_id
        out.setRequest request1.formatRequest(data_key, order_id, amount, intended_use)
        out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResIndRefundCC

The ResIndRefundCC will credit a specified amount to the cardholder's Credit Card. This transaction will process a regular Independent Refund on a card using the card information found in the Vault profile referenced by the data_key. Required fields for this transaction are: store_id, api_token, data_key, order_id, amount, and crypt_type. The optional variable is cust_id.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "resIndRef-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        amount = "1.00"
        crypt_type = "1"
        cust_id = "customer 2"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set request1 = server.CreateObject("Moneris.USResIndRefundCC")
            request1.setCustId cust_id
        out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResIndRefundAch

The ResIndRefundAch will credit a specified amount directly to the customer's bank account. Required fields for this transaction are: store_id, api_token, data_key, order_id, and amount. Optional variable is cust_id. This transaction will be processed as an us_ach_credit. The ach_info registered for this profile will be used. The details submitted within 'ach_info' will be returned in the response within ResolveData.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "resIndRef-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Mu0SL5tTX1t9VrCfl5G4PosZC"
        amount = "1.00"
        cust_id = "customer 3"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResIndRefundAch")
                request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

7. Financial Transactions with Extra Features - Examples

In the previous section the instructions were provided for the financial transaction set. eSELECTplus also provides several extra features/functionalities for the financial transactions. These features include storing customer and order details, verifying the card verification digit (CVD), verifying the address via address verification (AVS), and providing details for the Recurring Billing feature. AVS, CVD, and Recurring Billing must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated.

ResPurchaseCC (with Customer and Order details)

Below is an example of sending a ResPurchaseCC with the customer and order details. If one piece of CustInfo is sent then all fields must be included in the request. Unwanted fields need to be blank. Please see Appendix C. CustInfo Fields for description of each of the fields. It can be used in conjunction with other extra features such as AVS, CVD and Recurring Billing. ***Please note that the CustInfo fields are not used for any type of address verification or fraud check.***

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "resPreauth-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        amount = "1.00"
        crypt_type = "1"
        cust_id = "customer 2"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPreauthCC")
            billing=request1.formatAddress("bfname", "blname", "bCompany Name", "baddress", "bcity",
                       "bprovince", "bpostal", "bcountry", "bphone1", "bfax" , "btax1", "btax2", "btax3",
                       "bshipping_cost")
            shipping=request1.formatAddress("sfname", "slname", "sCompany Name", "saddress", "scity",
                       "sprovince", "spostal", "scountry", "sphone1", "sfax" , "stax1", "stax2", "stax3",
                       "sshipping_cost")
            request1.setCustInfo billing,shipping,"aa@email.com","take this instruction"
            request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"

        Response.Write "TransDate:  " & out.getTransDate & "<br>"
```

```
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

ResPurchaseAch (with Customer and Order details)

This transaction is processed as an us_ach_debit transaction with cust_info attached.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "preauth-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Mu0SL5tTX1t9VrCf15G4PosZC"
        amount = "1.00"
        cust_id = "customer 2"
        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPurchaseAch")
            billing=request1.formatAddress("bfname", "blname", "bCompany Name", "baddress", "bcity",
                 "bprovince", "bpostal", "bcountry", "bphone1", "bfax" , "btax1", "btax2", "btax3",
                 "bshipping_cost")
            shipping=request1.formatAddress("sfname", "slname", "sCompany Name", "saddress", "scity",
                 "sprovince", "spostal", "scountry", "sphone1", "sfax" , "stax1", "stax2", "stax3", "sshipping_cost")
            request1.setCustInfo billing,shipping,"aa@email.com","take this instruction"
            request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResPurchasePinless (with Customer and Order details)

This transaction will perform a us_pinless_debit_purchase with customer information added.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"
        order_id = "preauth-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "RJGjZ6ajb4mT9VNM6whYLZX4l"
        amount = "1.00"
        intended_use = "1"
        p_account_number = "234567899876543212345678 9"
        cust_id = "customer 2"
        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"
        Set request1 = server.CreateObject("Moneris.USResPurchasePinless")
            request1.setPAccountNumber p_account_number
            request1.setCustId cust_id
            billing=request1.formatAddress("bfname", "blname", "bCompany Name", "baddress", "bcity",
                 "bprovince", "bpostal", "bcountry", "bphone1", "bfax" , "btax1", "btax2", "btax3", "bshipping_cost")
            shipping=request1.formatAddress("sfname", "slname", "sCompany Name", "saddress", "scity",
                 "sprovince", "spostal", "scountry", "sphone1", "sfax" , "stax1", "stax2", "stax3", "sshipping_cost")
            request1.setCustInfo billing,shipping,"aa@email.com","take this instruction"
        out.setRequest request1.formatRequest(data_key, order_id, amount, intended_use)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

ResPurchaseCC (with Recurring Billing)

Recurring Billing is a feature that allows the transaction information to be sent once and then re-billed on a specified interval for a certain number of times. This is a feature commonly used for memberships, subscriptions, or any other charge that is re-billed on a regular basis. The transaction is split into two parts; the recur information and the transaction information. Please see Appendix D. Recur Fields for description of each of the fields. The optional customer and order details can be included in the transaction using the method outlined above – *ResPurchaseCC (with Customer and Order Details).* This transaction allows the merchant to use the data registered within a profile to setup a customer for recurring billing. Once a recurring billing transaction has been initiated, it will no longer be linked to the Vault profile. All changes to the recurring billing details will need to be made using the Recurring Billing tools, for example, using the Merchant Resource Centre. Recurring Billing must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
	store_id = "monusqa002"
	api_token = "qatoken"
	order_id = "resPurch-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
	data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
	amount = "1.00"
	crypt_type = "1"
	cust_id = "customer 2"

	recur_unit = "month"
	start_now = "true"
	start_date = "2011/11/01"
	num_recurs = "4"
	period = "1"
	recur_amount = "15.00"

	Set out = server.CreateObject("Moneris.USResolverRequest")
	out.initRequest store_id, api_token, "esplusqa.moneris.com"

	Set request1 = server.CreateObject("Moneris.USResPurchaseCC")
	request1.setRecur recur_unit,start_now,start_date,num_recurs,period,recur_amount
	request1.setCustId cust_id

	out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
	out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
	Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
	Response.Write "Response Code:  " & out.getResponseCode & "<br>"
	Response.Write "Transaction Type:  " & out.getTransType & "<br>"
	Response.Write "Message:  " & out.getMessage & "<br>"
	Response.Write "Amount:  " & out.getTransAmount & "<br>"
	Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
	Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
	Response.Write "Transaction ID:  " & out.getTransID & "<br>"
	Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
	Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
	Response.Write "Card Type:  " & out.getCardType & "<br>"
	Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
	Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
	Response.Write "TransDate:  " & out.getTransDate & "<br>"
	Response.Write "TransTime:  " & out.getTransTime & "<br>"
	Response.Write "Complete:  " & out.getComplete & "<br>"
	Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
	Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
	Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
```

```
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

As part of the Recurring Billing response there will be an additional method called getRecurSuccess. This can return a value of 'true' or 'false' based on whether the recurring transaction was successfully registered in our database.

ResPurchaseAch (with Recurring Billing)

This transaction is processed as an us_ach_debit with recur.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "resPurch-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Mu0SL5tTX1t9VrCfl5G4PosZC"
        amount = "1.00"
        cust_id = "customer 2"

        recur_unit = "month"
        start_now = "true"
        start_date = "2011/11/01"
        num_recurs = "4"
        period = "1"
        recur_amount = "15.00"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPurchaseAch")
        request1.setRecur recur_unit,start_now,start_date,num_recurs,period,recur_amount
        request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"

        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
```

```
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

As part of the Recurring Billing response there will be an additional method called getRecurSuccess( ). This can return a value of 'true' or 'false' based on whether the recurring transaction was successfully registered in our database.

ResPurchasePinless (with Recurring Billing)

This transaction will process a us_pinless_debit_purchase with recur.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "ResPurch-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "RJGjZ6ajb4mT9VNM6whYLZX4l"
        amount = "1.00"
        recur_unit = "month"
        start_now = "true"
        start_date = "2011/11/01"
        num_recurs = "4"
        period = "1"
        recur_amount = "15.00"
        cust_id = "customer 2"
        intended_use = "1"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPurchasePinless")
            request1.setPAccountNumber p_account_number
            request1.setCustId cust_id
            request1.setRecur recur_unit,start_now,start_date,num_recurs,period,recur_amount

        out.setRequest request1.formatRequest(data_key, order_id, amount, intended_use)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"

        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"

        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
```

```
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>
</BODY>
</HTML>
```

As part of the Recurring Billing response there will be an additional method called getRecurSuccess( ). This can return a value of 'true' or 'false' based on whether the recurring transaction was successfully registered in our database.

ResPurchaseCC (with CVD and AVS - eFraud)

Below is an example of a ResPurchaseCC transaction with CVD and AVS information. These values can be sent in conjunction with other additional variables such as Recurring Billing or customer information. It is important to note that if AVS details are sent, they will be submitted with the purchase but not stored. If they are not sent but avs_info is stored in the Vault profile, it will be submitted instead. If they are not sent and there was no stored avs_info found, no address verification will take place. To form CvdInfo please refer to Appendix I. Card Validation Digits (CVD) , to form AvsInfo please refer to Appendix J. Address Verification Service (AVS). To have the eFraud feature added to your profile, please call the Service Centre at 1-866-423-8475 to have your profile updated.

We strongly recommend that you include Address Verification (AVS) with all of your manually input transactions (MOTO/eCommerce). Doing so will ensure transactions are qualifying at the best possible interchange rate and will minimize costs to accept credit cards. If AVS is not present, the transaction may be assessed a higher interchange fee.

When testing AVS (eFraud) you **must only use** the Visa test card numbers, 4242424242424242 or 4005554444444403, and the amounts described in the Simulator eFraud Response Codes document available at https://developer.moneris.com

NOTE The CVD Value supplied by the cardholder should simply be passed to the eSelectPlus payment gateway. Under no circumstances should it be stored for subsequent uses or displayed as part of the receipt information.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = "monusqa002"
        api_token = "qatoken"

        order_id = "resPurch-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
        data_key = "Q6w3v99jgKs2snTocQmJjFxGc"
        amount = "10.42"
        crypt_type = "1"
        cust_id = "customer 2"

        avs_street_number = "6600"
        avs_street_name = "New York Street"
        avs_zipcode = "90210"

        cvd_indicator = "1"
        cvd_value = "333"

        Set out = server.CreateObject("Moneris.USResolverRequest")
        out.initRequest store_id, api_token, "esplusqa.moneris.com"

        Set request1 = server.CreateObject("Moneris.USResPurchaseCC")
            request1.setAvsInfo avs_street_number, avs_street_name, avs_zipcode
            request1.setCvdInfo cvd_indicator, cvd_value
            request1.setCustId cust_id

        out.setRequest request1.formatRequest(data_key, order_id, amount, crypt_type)
        out.sendRequest
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>

<%
        Response.Write "Receipt ID:  " & out.getReceiptID & "<br>"
        Response.Write "Response Code:  " & out.getResponseCode & "<br>"
```

```
        Response.Write "Transaction Type:  " & out.getTransType & "<br>"
        Response.Write "Message:  " & out.getMessage & "<br>"
        Response.Write "Amount:  " & out.getTransAmount & "<br>"
        Response.Write "AvsResultCode:  " & out.getAvsResultCode & "<br>"
        Response.Write "CVDResultCode:  " & out.getCvdResultCode & "<br>"
        Response.Write "Transaction ID:  " & out.getTransID & "<br>"
        Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
        Response.Write "CardLevelResult:  " & out.getCardLevelResult & "<br>"
        Response.Write "Card Type:  " & out.getCardType & "<br>"
        Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
        Response.Write "Recur Success:  " & out.getRecurSuccess & "<br>"
        Response.Write "TransDate:  " & out.getTransDate & "<br>"
        Response.Write "TransTime:  " & out.getTransTime & "<br>"
        Response.Write "Complete:  " & out.getComplete & "<br>"
        Response.Write "TimedOut:  " & out.getTimedOut & "<br>"
        Response.Write "ResSuccess:  " & out.getResSuccess & "<br>"
        Response.Write "PaymentType:  " & out.getPaymentType & "<br>"
        Response.Write "ResDataDataKey:  " & out.getDataKey & "<br>"
        Response.Write "ResDataPaymentType:  " & out.getResDataPaymentType & "<br>"
        Response.Write "ResDataCustId:  " & out.getResDataCustId & "<br>"
        Response.Write "ResDataPhone:  " & out.getResDataPhone & "<br>"
        Response.Write "ResDataEmail:  " & out.getResDataEmail & "<br>"
        Response.Write "ResDataNote:  " & out.getResDataNote & "<br>"
        Response.Write "ResDataMaskedPan:  " & out.getResDataMaskedPan & "<br>"
        Response.Write "ResDataExpDate:  " & out.getResDataExpDate & "<br>"
        Response.Write "Presentation Type:  " & out.getResDataPresentationType & "<br>"
        Response.Write "Account Number:  " & out.getResDataPAccountNumber & "<br>"
        Response.Write "ResDataCryptType:  " & out.getResDataCryptType & "<br>"
        Response.Write "Sec:  " & out.getResDataSec & "<br>"
        Response.Write "Cust First Name:  " & out.getResDataCustFirstName & "<br>"
        Response.Write "Cust Last Name:  " & out.getResDataLastName & "<br>"
        Response.Write "Cust Address 1:  " & out.getResDataCustAddress1 & "<br>"
        Response.Write "Cust Address 2:  " & out.getResDataCustAddress2 & "<br>"
        Response.Write "Cust City:  " & out.getResDataCustCity & "<br>"
        Response.Write "Cust State:  " & out.getResDataCustState & "<br>"
        Response.Write "Cust Zip:  " & out.getResDataCustZip & "<br>"
        Response.Write "Routing Num:  " & out.getResDataRoutingNum & "<br>"
        Response.Write "Masked Account Num:  " & out.getResDataMaskedAccountNum & "<br>"
        Response.Write "Check Num:  " & out.getResDataCheckNum & "<br>"
        Response.Write "Account Type:  " & out.getResDataAccountType & "<br>"
%>

</BODY>
</HTML>
```

As part of the eFraud response there will be two additional methods called getAvsResultCode( ) and getCvdResultCode( ). In the ResolveData, the AVS fields will be returned if avs_info is stored in the profile. If no avs_info was submitted with the purchase, then these details would have been used for verification. For a list of possible CVD responses please refer to Appendix I. Card Validation Digits (CVD) and for a list of AVS responses, please refer to Appendix J. Address Verification Service (AVS).

8. How Do I Test My Solution?

A testing environment is available for you to connect to while you are integrating your site to our payment gateway. 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 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 APIs in the test environment you will need to use 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**
monusqa002*	qatoken	demouser	abc1234
monusqa003	qatoken	demouser	abc1234
monusqa004	qatoken	demouser	abc1234
monusqa005	qatoken	demouser	abc1234
monusqa006	qatoken	demouser	abc1234

* test store 'monusqa002' is intended for testing the Pinless Debit transactions

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

Test Card Numbers	
Card Plan	**Card Number**
MasterCard	5454545454545454
Visa	4242424242424242 or 4005554444444403
Amex	373599005095005
Pinless Debit	4496270000164824

Test bank Account Details			
Financial Institution	**Routing Number**	**Account Number**	**Check Number**
FEDERAL RESERVE BANK	011000015	Any number between 5-22 digits	Any number

To access the Merchant Resource Centre in the test environment go to https://esplusqa.moneris.com/usmpg. 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 credit card 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 $11.00. 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 https://developer.moneris.com


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

The test environment will approve/register all ACH transactions as long as there is no error with the format. For example, if all of the ACH variables are properly named and populated, all transactions will approve/register. If there is a format violation, such as invalid data in one of the fields (ex. cust_zip requires 'MI' but 'Michigan' is sent) then the ACH transaction will decline/fail to register.

9. How Do I Get Help?

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

For technical support:
Phone: 1-866-696-0488

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

When sending an email support request please be sure to mention that this is in reference to a Vault transaction, your name, 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.**

10. Appendix A. Definition of Request Fields

Request Fields		
Variable Name	**Size/Type**	**Description**
order_id	50 / an	Merchant defined unique transaction identifier - must be unique for every ResPurchase, ResPreAuth and ResIndRefund attempt. Characters allowed for Order ID: **a-z A-Z 0-9 _ - : . @ spaces**
data_key	23 / an	An alphanumeric identifier used in Vault transactions to uniquely identify a Vault profile. The data_key is generated by Moneris Solutions and returned to the merchant when the profile is first registered using ResAddCC, ResAddACH or ResAddPinless transactions.
pan	20 / variable	Credit or Pinless Debit Card Number - no spaces or dashes. Most credit/pinless debit 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
enc_track2		This is a string that is retrieved by swiping or keying in a credit card through a Moneris provided encrypted mag swipe card reader. It is part of an encrypted keyed or swiped transaction only. This string must be retrieved by a specific device. Please refer to device_type for the list of current available devices.
device_type	an	Defines the encrypted mag swipe reader that was used for swiping or keying in the credit card. Plesase note, this device must be provided by Moneris Solutions so that the values are properly encrypted and decrypted. This field is case sensitive. Available values are: device_type="idtech"
amount	9 / decimal	Amount of the transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
crypt_type	1 / an	E-Commerce Indicator: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 7 - SSL enabled merchant 8 - Non Secure Transaction (Web or Email Based) 9 - SET non - Authenticated transaction
cust_id	50 / an	This is an optional field that can be either registered in a profile or sent as part of a ResPurchase, ResPreauth or ResIndRefund request. It is searchable from the Moneris Merchant Resource Centre. It is commonly used for policy number, membership number, student ID or invoice number.
phone	30 / an	Phone number of the customer. This is an optional field which can be sent in when creating or updating a Vault profile.
email	30 / an	Email of the customer. This is an optional field which can be sent in when creating or updating a Vault profile.

note	30 / an	This field can be used for supplementary information which is to be sent in with the transaction. This is an optional field which can be sent in when creating or updating a Vault profile.
intended_use	1 / num	Identifies the party who initiated the transaction. - "0" = Merchant initiated the payment - "1" = Customer initiated the payment
p_account_number	25 / an	The billing invoice number – no spaces or dashes. The length of the account number varies with a maximum length of 25 digits. This is field is mandatory to properly process a Pinless Debit financial transaction. It must either be registered in the profile or submitted at the time of the ResPurchasePinless transaction.
presentation_type	1 / alpha	Identifies how merchants obtain the Pinless Debit account. This field is a mandatory field required when adding the Pinless Debit profile. - 'X' for Telephone/VRU - 'W' for Internet
avs_street_number avs_street_name	19 / an	Street Number & Street Name (max – 19 digit limit for street number and street name combined). This must match the address that the issuing bank has on file.
avs_zipcode	9 / an	Zip or Postal Code – This must match what the issuing bank has on file.
cvd_value	4 / num	Credit Card CVD value – this number accommodates either 3 or 4 digit CVD values. Refer to Appendix G for further details. Note: The CVD value supplied by the cardholder should simply be passed to the eSELECTplus payment gateway. Under no circumstances should it be stored for subsequent uses or displayed as part of the receipt information.
cvd_indicator	1 / num	CVD presence indicator (1 digit – refer to Appendix G for values)
commcard_invoice	17 / an	Level 2 Invoice Number for the transaction. Used for Corporate Credit Card transactions (Commercial Purchasing Cards). Characters allowed for commcard_invoice: **a-z A-Z 0-9 spaces**
commcard_tax_amount	9 / decimal	Level 2 Tax Amount of the transaction. Used for Corporate Credit Card transactions (Commercial Purchasing Cards). This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum is 9999999.99
dynamic_descriptor	25 / an	Merchant defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name. Please note, the combined length of the merchant's business name and dynamic_descriptor may not exceed 25 characters.

NOTE

The order_id allows the following characters: **a-z A-Z 0-9 _ - : . @ spaces**

The commcard_invoice allows the following characters: **a-z A-Z 0-9 spaces**

All other request fields allow the following characters: **a-z A-Z 0-9 _ - : . @ $ =** /

11. Appendix B. Definitions of Response Fields

Response Fields		
Variable Name	**Size/Type**	**Description**
ReceiptId	50 / an	order_id specified in request
ReferenceNum	18 / num	The reference number is an 18 character string that references the terminal used to process the transaction as well as the shift, batch and sequence number, This data is typically used to reference transactions on the host systems and must be displayed on any receipt presented to the customer. This information should be stored by the merchant. The following illustrates the breakdown of this field where "640123450010690030" is the reference number returned in the message, "64012345" is the terminal id, "001" is the shift number, "069" is the batch number and "003" is the transaction number within the batch. Moneris Host Transaction identifier.
ReponseCode	3 / num	Transaction Response Code Financial Transaction Responses (i.e. ResPurchase) < 50 Transaction approved >= 50 Transaction declined NULL Transaction was not sent for authorization * If you would like further details on the response codes that are returned please see the Response Codes document available at https://developer.moneris.com Vault Admin Responses (i.e. ResAdd or ResDelete) 001 Successfully registered (CC|ACH|Pinless) details. Successfully updated (CC|ACH|Pinless) details. Successfully deleted (CC|ACH|Pinless) details. Successfully located (CC|ACH|Pinless) details. Successfully located # expiring cards. (NOTE: # = the number of cards located) 983 Can not find previous 986 Incomplete: timed out 987 Invalid transaction 988 Can not find expiring cards Null Error: Malformed XML
AuthCode	8 / an	Authorization code returned from the issuing institution
TransTime	##:##:##	Processing host time stamp
TransDate	yyyy-mm-dd	Processing host date stamp
TransType	an	Type of transaction that was performed
Complete	true/false	Transaction was sent to authorization host and a response was received
Message	100 / an	Response description returned from issuing institution.
TransAmount		
CardType	2 / alpha	Credit Card Type
Txn_number	20 / an	Gateway Transaction identifier
TimedOut	true/false	Transaction failed due to a process timing out
Ticket	n/a	reserved
RecurSuccess	true/false	Indicates whether the recurring billing transaction successfully registered.
AvsResultCode	1/alpha	Indicates the address verification result. Refer to Appendix H.

CvdResultCode	2/an	Indicates the CVD validation result. Refer to Appendix G.
ResSuccess	true/false	Indicates if Vault transaction was successful.
PaymentType	cc|ach|pinless	Indicates the payment type associated with a Vault profile.
DataKey	23 / an	The data_key specified in the request. If processing a ResAdd transaction, then this will indicate the newly generated unique data_key associated with the new profile.
ResolveData		The fields returned within ResolveData will coincide with the registered profile details. Please refer to the examples. Fields found in ResolveData are: data_key, payment_type, cust_id, phone, email, note, masked_pan, pan, expdate, crypt_type, avs_street_number, avs_street_name, avs_zipcode, presentation_type, p_account_number, sec, cust_first_name, cust_last_name, cust_address1, cust_address2, cust_city, cust_state, cust_zip, routing_num, masked_account_num, account_num, check_num, and account_type.

12. Appendix C. CustInfo Fields

Field Definitions		
Field Name	**Size/Type**	**Description**
Billing and Shipping Information		
NOTE: The fields for billing and shipping information are identical. Please refer to section 7 for an example.		
first_name	30 / an	
last_name	30 / an	
company_name	30 / an	
address	30 / an	
city	30 / an	
province	30 / an	
postal_code	30 / an	
country	30 / an	
phone	30 / an	
fax	30 / an	
tax1	30 / an	
tax2	30 / an	
tax3	30 / an	
shipping_cost	30 / an	
Item Information		
NOTE: You may send multiple items - please refer to section 7 for an example.		
item_description	30 / an	
item_quantity	10 / num	You must send a quantity > 0 or the item will not be added to the item list (ie. minimum 1, maximum 9999999999)
item_product_code	30 / an	
item_extended_amount	9 /decimal	This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
Extra Details		
email	50 / an	
instructions	50 / an	

	If you send characters that are not included in the allowed list, these extra transaction details may not be stored.
NOTE	All fields are alphanumeric and allow the following characters: **a-z A-Z 0-9 _ - : . @ $ =** /
	Also, the data sent in Billing and Shipping Address fields will not be used for any address verification. Please refer to the section 7 for further details about Address Verification Service (AVS).

13. Appendix D. Recur Fields

Recur Request Fields		
Variable Name	**Size/Type**	**Description**
recur_unit	day, week, month, eom	The unit that you wish to use as a basis for the Interval. This can be set as day, week, month or end of month. Then using the "period" field you can configure how many days, weeks, months between billing cycles.
period	0 – 999 / num	This is the number of recur_units you wish to pass between billing cycles. Example : period = 45, recur_unit=day -> Card will be billed every 45 days. period = 4, recur_unit=weeks -> Card will be billed every 4 weeks. period = 3, recur_unit=month -> Card will be billed every 3 months. period = 3, recur_unit=eom -> Card will be billed every 3 months (on the last day of the month) Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future.
start_date	YYYY/MM/DD	This is the date on which the first charge will be billed. The value must be in the future. It cannot be the day on which the transaction is being sent. If the transaction is to be billed immediately the start_now feature must be set to true and the start_date should be set at the desired interval after today.
start_now	true / false	When a charge is to be made against the card immediately start_now should be set to 'true'. If the billing is to start in the future then this value is to be set to 'false'. When start_now is set to 'true' the amount to be billed immediately may differ from the recur amount billed on a regular basis thereafter.
recur_amount	9 / decimal	Amount of the recurring transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99. This is the amount that will be billed on the start_date and every interval thereafter.
num_recurs	1 – 99 / num	The number of times to recur the transaction.
amount	9 / decimal	When start_now is set to 'true' the amount field in the transaction array becomes the amount to be billed immediately. When start_now is set to 'false' the amount field in the transaction array should be the same as the recur_amount field.

Recur Request Examples	
Recur Request Exampl	**Description**

`order_id = "monthly_purchase"`
`data_key = "3ixBzo1e25Zck8urjLcTbuu22"`
`amount = "15.00"`
`crypt_type = "7"`
 	
`recur_unit = "month"`
`start_now = "false"`
`start_date = "2007/01/02"`
`num_recurs = "12"`
`period = "2"`
`recur_amount = "30.00"`
 	
`request1.setRecur ecur_unit, start_now,`
`start_date, num_recurs, period, recur_amount`
	In the example to the left the first transaction will occur in the future on Jan 2nd 2007. It will be billed $30.00 every 2 months on the 2nd of each month. The card will be billed a total of 12 times.

`order_id = "bi-weekly_purchase"`
`data_key = "3ixBzo1e25Zck8urjLcTbuu22"`
`amount = "15.00"`
`crypt_type = "7"`
 	
`recur_unit = "week"`
`start_now = "true"`
`start_date = "2007/01/02"`
`num_recurs = "26"`
`period = "2"`
`recur_amount = "30.00"`
 	
`request1.setRecur ecur_unit, start_now,`
`start_date, num_recurs, period, recur_amount`
	In the example on the left the first charge will be billed immediately. The initial charge will be for $15.00. Then starting on Jan 2nd 2007 the credit card will be billed $30.00 every 2 weeks for 26 recurring charges. The card will be billed a total of 27 times. (1 x $15.00 (immediate) and 26 x $30.00 (recurring))

NOTE When completing the recurring billing portion please keep in mind that to prevent the shifting of recur bill dates, avoid setting the start_date for anything past the 28th of any given month. For example, all billing dates set for the 31st of May will shift and bill on the 30th in June and will then bill the cardholder on the 30th for every subsequent month.

14. Appendix E. AchInfo Fields

AchInfo Request Fields		
Variable Name	**Size/Type**	**Description**
sec	3 / an	ACH SEC Code: ppd - Prearranged Payment and Deposit ccd - Cash Concentration or Disbursement web - Internet Initiated Entry * only PPD and CCD apply to the ResIndRefundAch transaction
routing_num	9 / num	The first number in the MICR, or magnetic ink character recognition, line at the bottom of a check is the bank's check routing number. It is exactly nine digits long and always starts with 0, 1, 2 or 3.
account_num	50 / num	The account number may appear before or after the check number in the check's MICR line at the bottom of the check. The length of the account number varies with a maximum length of 50 digits.
check_num	16 / num	The sequential number for checks appears in both the MICR line at the bottom of the check and the upper right corner of the check. The check number length may vary; the maximum length is 16 digits. This is an optional field.
account_type	savings / checking	Identifies the type of bank account. The account type must be submitted as either 'savings' or 'checking'. This field is case sensitive.
ACH Customer Information		
NOTE: The following Account Holder information fields are optional.		
cust_first_name	50 / an	
cust_last_name	50 / an	
cust_address1	50 / an	
cust_address2	50 / an	
cust_city	50 / an	
cust_state	2 / alpha	The state must be submitted as exactly 2 characters (ex. MI – Michigan)
cust_zip	15 / an	

NOTE

If you send characters that are not included in the allowed list, the ACH transaction may not be properly registered.

All alphanumeric fields allow the following characters: **a-z A-Z 0-9 _ - : . @ $ =** /

Also, the data sent in the ACH Customer Information fields will not be used for any address verification.

15. Appendix F. ProgIDs

Request Object	Moneris.USRequest

The following 16 are ProgIDs for request formatters.

USResAddAchRequest	Moneris.USResAddAch
USResAddCCRequest	Moneris.USResAddCC
USEncResAddCCRequest	Moneris.USEncResAddCC
USResTokenizeCCRequest	Moneris.USResTokenizeCC
USResAddPinlessRequest	Moneris.USResAddPinless
USResDeleteRequest	Moneris.USResDelete
USResGetExpiringRequest	Moneris.USResGetExpiring
USResIndRefundAchRequest	Moneris.USResIndRefundAch
USResIndRefundCCRequest	Moneris.USResIndRefundCC
USResLookupFullRequest	Moneris.USResLookupFull
USResLookupMaskedRequest	Moneris.USResLookupMasked
USResPreauthCCRequest	Moneris.USResPreauthCC
USResPurchaseAchRequest	Moneris.USResPurchaseAch
USResPurchaseCCRequest	Moneris.USResPurchaseCC
USResPurchasePinlessRequest	Moneris.USResPurchasePinless
USResUpdateAchRequest	Moneris.USResUpdateAch
USResUpdateCCRequest	Moneris.USResUpdateCC
USEncResUpdateCCRequest	Moneris.USEncResUpdateCC
USResUpdatePinlessRequest	Moneris.USResUpdatePinless

16. Appendix G. Method List

F.1 MonerisUSRequest

```
HRESULT sendRequest(void);
HRESULT initRequest([in] BSTR store_id, [in] BSTR api_token, [in] BSTR gateway_url);
HRESULT setRequest([in] BSTR request);
HRESULT dumpXMLResponse([out,retval] BSTR* xmlresponse);
HRESULT getRecurSuccess([out,retval] BSTR* recursuccess);
HRESULT getReceiptID([out,retval] BSTR* receiptid);
HRESULT getReferenceNum([out,retval] BSTR* referencenum);
HRESULT getResponseCode([out,retval] BSTR* responsecode);
HRESULT getISO([out,retval] BSTR* iso);
HRESULT getAuthCode([out,retval] BSTR* authcode);
HRESULT getTransTime([out,retval] BSTR* transtime);
HRESULT getTransDate([out,retval] BSTR* transdate);
HRESULT getTransType([out,retval] BSTR* transtype);
HRESULT getCompleteStatus([out,retval] BSTR* complete);
HRESULT getMessage([out,retval] BSTR* message);
HRESULT getTransAmount([out,retval] BSTR* amount);
HRESULT getCardType([out,retval] BSTR* cardtype);
HRESULT getTransID([out,retval] BSTR* transid);
HRESULT getTimedOutStatus([out,retval] BSTR* timedout);
HRESULT getBankTotals([out,retval] BSTR* banktotals);
HRESULT getTicket([out,retval] BSTR* ticket);
HRESULT getAvsResultCode([out,retval] BSTR* AvsResultCode);
HRESULT getCvdResultCode([out,retval] BSTR* CvdResultCode);
HRESULT dumpXMLRequest([out,retval] BSTR* xmlrequest);
HRESULT dumpURL([out,retval] BSTR* url);
HRESULT getMPIPAResVerified([in] BSTR* PAResVerified);
HRESULT getMPIMessage([out,retval] BSTR* message);
```

```
HRESULT getMPISuccess([out,retval] BSTR* success);
HRESULT getMPIType([out,retval] BSTR* type);
HRESULT initMpiRequest([in] BSTR gateway_url);
HRESULT getMPIInlineForm([out,retval] BSTR* popupform);
HRESULT getMPITermUrl([out,retval] BSTR* termurl);
HRESULT getMPIAcsUrl([out,retval] BSTR* acsurl);
HRESULT getMPICavv([out,retval] BSTR* cavv);
HRESULT getMPIPaReq([out,retval] BSTR* pareq);
HRESULT getMPIMd([out,retval] BSTR* md);
```

F.2 Moneris.USResAddAch

```
HRESULT formatRequest([in] BSTR cust_id,
                      [in] BSTR phone,
                      [in] BSTR email,
                      [in] BSTR note,
                      [out,retval] BSTR* xmlrequest);
HRESULT setAchInfo([in] BSTR sec,
                   [in] BSTR cust_first_name,
                   [in] BSTR cust_last_name,
                   [in] BSTR cust_address1,
                   [in] BSTR cust_address2,
                   [in] BSTR cust_city,
                   [in] BSTR cust_state,
                   [in] BSTR cust_zip,
                   [in] BSTR routing_num,
                   [in] BSTR account_num,
                   [in] BSTR check_num,
                   [in] BSTR account_type);
```

F.3 Moneris.USResAddCC

```
HRESULT formatRequest([in] BSTR cust_id,
                      [in] BSTR phone,
                      [in] BSTR email,
                      [in] BSTR note,
                      [in] BSTR pan,
                      [in] BSTR expdate,
                      [in] BSTR crypt_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
```

F.4 Moneris.USEncResAddCC

```
HRESULT formatRequest([in] BSTR cust_id,
                      [in] BSTR phone,
                      [in] BSTR email,
                      [in] BSTR note,
                      [in] BSTR enc_track2,
                      [in] BSTR device_type,
                      [in] BSTR crypt_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
```

F.5 Moneris.USResTokenizeCC

```
HRESULT formatRequest([in] BSTR order_id,
                      [in] BSTR txn_number,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPhone([in] BSTR phone);
HRESULT setEmail([in] BSTR email);
HRESULT setNote([in] BSTR note);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
```

F.6 Moneris.USResAddPinless

```
HRESULT formatRequest([in] BSTR cust_id,
                      [in] BSTR phone,
                      [in] BSTR email,
                      [in] BSTR note,
                      [in] BSTR pan,
                      [in] BSTR presentation_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setExpdate([in] BSTR expdate);
HRESULT setPAccountNumber([in] BSTR p_account_number);
```

F.7 Moneris.USResDelete

```
HRESULT formatRequest([in] BSTR data_key, [out,retval] BSTR* xmlrequest);
```

F.8 Moneris.USResGetExpiring

```
HRESULT formatRequest([out,retval] BSTR* xmlrequest);
```

F.9 Moneris.USResIndRefundAch

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
```

F.10 Moneris.USResIndRefundCC

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [in] BSTR crypt_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
```

F.11 Moneris.USResLookupFull

```
HRESULT formatRequest([in] BSTR data_key,
                      [out,retval] BSTR* xmlrequest);
```

F.12 Moneris.USResLookupMasked

```
HRESULT formatRequest([in] BSTR data_key,
[out,retval] BSTR* xmlrequest);
```

F.13 Moneris.USResPreauthCC

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [in] BSTR crypt_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setCvdInfo([in] BSTR cvd_indicator,
                   [in] BSTR cvd_value);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
HRESULT formatAddress([in] BSTR first_name,
                      [in] BSTR last_name,
                      [in] BSTR company_name,
                      [in] BSTR address,
                      [in] BSTR city,
                      [in] BSTR province,
                      [in] BSTR postal_code,
                      [in] BSTR country,
                      [in] BSTR phone_number,
                      [in] BSTR fax,
                      [in] BSTR tax1,
                      [in] BSTR tax2,
                      [in] BSTR tax3,
                      [in] BSTR shipping_cost,
                      [out,retval] BSTR* xmladdress);
HRESULT setItem([in] BSTR name,
                [in] BSTR quantity,
                [in] BSTR product_code,
                [in] BSTR extended_amount);
HRESULT setCustInfo([in] BSTR billing,
                    [in] BSTR shipping,
                    [in] BSTR email,
                    [in] BSTR instructions);
```

F.14 Moneris.USResPurchaseAch

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT formatAddress([in] BSTR first_name,
                      [in] BSTR last_name,
                      [in] BSTR company_name,
                      [in] BSTR address,
                      [in] BSTR city,
                      [in] BSTR province,
                      [in] BSTR postal_code,
                      [in] BSTR country,
                      [in] BSTR phone_number,
                      [in] BSTR fax,
                      [in] BSTR tax1,
                      [in] BSTR tax2,
                      [in] BSTR tax3,
                      [in] BSTR shipping_cost,
                      [out,retval] BSTR* xmladdress);
```

```
HRESULT setItem([in] BSTR name,
                [in] BSTR quantity,
                [in] BSTR product_code,
                [in] BSTR extended_amount);
HRESULT setCustInfo([in] BSTR billing,
                    [in] BSTR shipping,
                    [in] BSTR email,
                    [in] BSTR instructions);
HRESULT setRecur([in] BSTR recur_unit,
                [in] BSTR start_now,
                [in] BSTR start_date,
                [in] BSTR num_recurs,
                [in] BSTR period,
                [in] BSTR recur_amount);
```

F.15 Moneris.USResPurchaseCC

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [in] BSTR crypt_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setCommcardInvoice([in] BSTR commcard_invoice);
HRESULT setCommcardTaxAmount([in] BSTR commcard_tax_amount);
HRESULT setCvdInfo([in] BSTR cvd_indicator,
                   [in] BSTR cvd_value);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
HRESULT formatAddress([in] BSTR first_name,
                      [in] BSTR last_name,
                      [in] BSTR company_name,
                      [in] BSTR address,
                      [in] BSTR city,
                      [in] BSTR province,
                      [in] BSTR postal_code,
                      [in] BSTR country,
                      [in] BSTR phone_number,
                      [in] BSTR fax,
                      [in] BSTR tax1,
                      [in] BSTR tax2,
                      [in] BSTR tax3,
                      [in] BSTR shipping_cost,
                      [out,retval] BSTR* xmladdress);
HRESULT setItem([in] BSTR name,
                [in] BSTR quantity,
                [in] BSTR product_code,
                [in] BSTR extended_amount);
HRESULT setCustInfo([in] BSTR billing,
                    [in] BSTR shipping,
                    [in] BSTR email,
                    [in] BSTR instructions);
HRESULT setRecur([in] BSTR recur_unit,
                [in] BSTR start_now,
                [in] BSTR start_date,
                [in] BSTR num_recurs,
                [in] BSTR period,
```

```
                    [in] BSTR recur_amount);
```

F.16 Moneris.USResPurchasePinless

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR order_id,
                      [in] BSTR amount,
                      [in] BSTR intended_use,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPAccountNumber([in] BSTR p_account_number);
HRESULT formatAddress([in] BSTR first_name,
                      [in] BSTR last_name,
                      [in] BSTR company_name,
                      [in] BSTR address,
                      [in] BSTR city,
                      [in] BSTR province,
                      [in] BSTR postal_code,
                      [in] BSTR country,
                      [in] BSTR phone_number,
                      [in] BSTR fax,
                      [in] BSTR tax1,
                      [in] BSTR tax2,
                      [in] BSTR tax3,
                      [in] BSTR shipping_cost,
                      [out,retval] BSTR* xmladdress);
HRESULT setItem([in] BSTR name,
                [in] BSTR quantity,
                [in] BSTR product_code,
                [in] BSTR extended_amount);
HRESULT setCustInfo([in] BSTR billing,
                    [in] BSTR shipping,
                    [in] BSTR email,
                    [in] BSTR instructions);
HRESULT setRecur([in] BSTR recur_unit,
                 [in] BSTR start_now,
                 [in] BSTR start_date,
                 [in] BSTR num_recurs,
                 [in] BSTR period,
                 [in] BSTR recur_amount);
```

F.17 Moneris.USResUpdateAch

```
HRESULT formatRequest([in] BSTR data_key,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPhone([in] BSTR phone);
HRESULT setEmail([in] BSTR email);
HRESULT setNote([in] BSTR note);
HRESULT setSec([in] BSTR sec);
HRESULT setCustFirstName([in] BSTR cust_first_name);
HRESULT setCustLastName([in] BSTR cust_last_name);
HRESULT setCustAddress1([in] BSTR cust_address1);
HRESULT setCustAddress2([in] BSTR cust_address2);
HRESULT setCustCity([in] BSTR cust_city);
HRESULT setCustState([in] BSTR cust_state);
HRESULT setCustZip([in] BSTR cust_zip);
HRESULT setRoutingNum([in] BSTR routing_num);
HRESULT setAccountNum([in] BSTR account_num);
```

```
HRESULT setCheckNum([in] BSTR check_num);
HRESULT setAccountType([in] BSTR account_type);
HRESULT setAchInfo([in] BSTR sec,
                   [in] BSTR cust_first_name,
                   [in] BSTR cust_last_name,
                   [in] BSTR cust_address1,
                   [in] BSTR cust_address2,
                   [in] BSTR cust_city,
                   [in] BSTR cust_state,
                   [in] BSTR cust_zip,
                   [in] BSTR routing_num,
                   [in] BSTR account_num,
                   [in] BSTR check_num,
                   [in] BSTR account_type);
```

F.18 Moneris.USEncResUpdateCC

```
HRESULT formatRequest([in] BSTR data_key,
                      [in] BSTR enc_track2,
                      [in] BSTR device_type,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPhone([in] BSTR phone);
HRESULT setEmail([in] BSTR email);
HRESULT setNote([in] BSTR note);
HRESULT setCryptType([in] BSTR crypt_type);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
HRESULT setAvsStreetName([in] BSTR avs_street_name);
HRESULT setAvsStreetNumber([in] BSTR avs_street_number);
HRESULT setAvsZipcode([in] BSTR avs_zipcode);
```

F.19 Moneris.USResUpdateCC

```
HRESULT formatRequest([in] BSTR data_key,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPhone([in] BSTR phone);
HRESULT setEmail([in] BSTR email);
HRESULT setNote([in] BSTR note);
HRESULT setPan([in] BSTR pan);
HRESULT setExpdate([in] BSTR expdate);
HRESULT setCryptType([in] BSTR crypt_type);
HRESULT setAvsInfo([in] BSTR avs_street_number,
                   [in] BSTR avs_street_name,
                   [in] BSTR avs_zipcode);
HRESULT setAvsStreetName([in] BSTR avs_street_name);
HRESULT setAvsStreetNumber([in] BSTR avs_street_number);
HRESULT setAvsZipcode([in] BSTR avs_zipcode);
```

F.20 Moneris.USResUpdatePinless

```
HRESULT formatRequest([in] BSTR data_key,
                      [out,retval] BSTR* xmlrequest);
HRESULT setCustId([in] BSTR cust_id);
HRESULT setPhone([in] BSTR phone);
HRESULT setEmail([in] BSTR email);
HRESULT setNote([in] BSTR note);
HRESULT setPan([in] BSTR pan);
HRESULT setExpdate([in] BSTR expdate);
HRESULT setPresentationType([in] BSTR presentation_type);
HRESULT setPAccountNumber([in] BSTR p_account_number);
```

17. Appendix H. Error Messages

E_GATEWAY_CONNECTION	0x80040407	-2147220473
E_BAD_GATEWAY_URL	0x8004040d	-2147220467
E_UNABLE_TO_PARSE_CONFIG	0x80040411	-2147220463
E_CANNOT_GET_HSESSION	0x80040412	-2147220462
E_CANNOT_GET_HCONNECTION	0x80040413	-2147220461
E_CANNOT_GET_HRESULTS	0x80040414	-2147220460
E_HEADER_PROBLEM	0x80040415	-2147220459
E_OPTION_PROBLEM	0x80040416	-2147220458

Response Code = NULL – The response code can be returned as null for a variety of reasons. A majority of the of the time the explanation is contained within the Message field. When a 'NULL' response is returned it can indicate that the Issuer, the credit card host, or the gateway is unavailable, either because they are offline or you are unable to connect to the internet. A 'NULL' can also be returned when a transaction message is improperly formatted.

Vault Specific Responses

Message: Can not find previous
Cause: data_key provided was not found in our records or profile is no longer active.

Message: Invalid Transaction
Cause: -Transaction can not be performed due to improper data being sent in.
-Mandatory field is missing or an invalid SEC code is sent in.

Message: Malformed XML
Cause: Parse error.

Message: Incomplete
Cause: -Timed out.
-Can not find expiring cards.

18. Appendix I. Card Validation Digits (CVD)

The Card Validation Digits (CVD) value refers to the numbers appearing on the back of the credit card which are not imprinted on the front. The exception to this is with American Express cards where this value is indeed printed on the front. The mpgCvdInfo parameter is broken down into two elements. The first element is the CVD Value itself.

The second element is the CVD Indicator. This value indicates the possible scenarios when collecting CVD information. This is a 1 digit value which can have any of the following values:

CVD INDICATOR	
VALUE	**DEFINITION**
0	CVD value is deliberately bypassed or is not provided by the merchant.
1	CVD value is present.
2	CVD value is on the card, but is illegible.
9	Cardholder states that the card has no CVD imprint.

CVD Response codes:

The CVD response is an alphanumeric 2 byte variable. The first byte is the numeric CVD indicator sent in the request; the second byte would be the response code. The following is a list of all possible responses once a CVD value has been passed in.

CVD RESPONSE CODES	
RESULT VALUE	**DEFINITION**
M	Match
Y	Match for AmEx
N	No Match
P	Not Processed
S	CVD should be on the card, but Merchant has indicated that CVD is not present
R	Retry for AmEx
U	Issuer is not a CVD participant
Other	Invalid Response Code


NOTE

The CVD value supplied by the cardholder should simply be passed to the eSELECTplus payment gateway. Under no circumstances should it be stored for subsequent uses or displayed as part of the receipt information.

***For additional information on how to handle these responses, please refer to Appendix I.**

19. Appendix J. Address Verification Service (AVS)

The Address Verification Service (AVS) value refers to the cardholder's street number, street name and zip/postal code as it would appear on their statement. mpgAvsInfo is broken down into three elements:

Element	Type	Length
Street Number	Numeric	19 characters combined.
Street Name	Alphanumeric	
Zip/Postal Code	Alphanumeric	9 characters

The following table outlines the possible responses when passing in AVS information.

AVS RESPONSE CODES		
VALUE	**VISA/DISCOVER / JCB**	**MASTERCARD**
A	Address matches, ZIP does not. Acquirer rights not implied.	Address matches, zip code does not.
B	Street addresses match. Zip code not verified due to incompatible formats. (Acquirer sent both street address and zip code.)	N/A
C	Street addresses not verified due to incompatible formats. (Acquirer sent both street address and zip code.)	N/A
D	Street addresses and zip codes match.	N/A
F	Street address and zip code match. Applies to U.K. only	N/A
G	Address information not verified for international transaction. Issuer is not an AVS participant, or AVS data was present in the request but issuer did not return an AVS result, or Visa performs AVS on behalf of the issuer and there was no address record on file for this account.	N/A
I	Address information not verified.	N/A
K	N/A	N/A
L	N/A	N/A
M	Street address and zip code match.	N/A
N	No match. Acquirer sent postal/ZIP code only, or street address only, or both zip code and street address. Also used when acquirer requests AVS but sends no AVS data.	Neither address nor zip code matches.
O	N/A	N/A
P	Zip code match. Acquirer sent both zip code and street address but street address not verified due to incompatible formats.	N/A
R	Retry: system unavailable or timed out. Issuer ordinarily performs AVS but was unavailable. The code R is used by Visa when issuers are unavailable. Issuers should refrain from using this code.	Retry; system unable to process.
S	N/A	AVS currently not supported.
U	Address not verified for domestic transaction. Issuer is not an AVS participant, or AVS data was present in the request but issuer did not return an AVS result, or Visa performs AVS on behalf of the issuer and there was no address record on file for this account.	No data from Issuer/Authorization system.
W	Not applicable. If present, replaced with 'Z' by Visa. Available for U.S. issuers only.	For U.S. Addresses, nine-digit zip code matches, address does not; for address outside the U.S. postal code matches, address does not.
X	N/A	For U.S. addresses, nine-digit zip code and addresses matches; for addresses outside the U.S., postal code and address match.
Y	Street address and zip code match.	For U.S. addresses, five-digit zip code and address matches.
Z	Postal/Zip matches; street address does not match or street address not included in request.	For U.S. addresses, five digit zip code matches, address does not.

VALUE	AMERICAN EXPRESS
A	Billing address matches, zip code does not
D	Customer name incorrect, zip code matches
E	Customer name incorrect, billing address and zip code match
F	Customer name incorrect, billing address matches
K	Customer name matches
L	Customer name and zip code match
M	Customer name, billing address, and zip code match
N	Billing address and zip code do not match
O	Customer name and billing address match
R	System unavailable; retry
S	AVS not currently supported
U	Information is unavailable
W	Customer name, billing address, and zip code are all incorrect
Y	Billing address and zip code both match
Z	Zip code matches, billing address does not

20. Appendix K. Additional Information for CVD and AVS

The responses that are received from CVD and AVS verifications are intended to provide added security and fraud prevention, but the response itself will not affect the completion of a transaction. Upon receiving a response, the choice to proceed with a transaction is left entirely to the merchant.

Please note that all responses coming back from these verification methods are not direct indicators of whether a merchant should complete any particular transaction. The responses should not be used as a strict guideline of which transaction will approve or decline.

NOTE Please note that CVD and AVS verification is only applicable towards Visa, MasterCard, Discover, JCB and American Express transactions.

21. Appendix L. Vault Receipts

When completing Vault financial transactions (ResPreauth, ResPurchase, ResIndRefund), a receipt will need to be presented to the customer. Receipt requirements depend on the type of transaction which was performed and the form of payment used (i.e. Credit Card vs ACH). For further details on all receipt requirements, please refer to the full COM API Integration Guide available at: https://developer.moneris.com

eSELECTplus™

Copyright Notice

This document has been produced as a reference guide to assist Moneris client's hereafter referred to as merchants. Every effort has been made to the make the information in this reference guide as accurate as possible. The authors of Moneris Solutions shall have neither liability nor responsibility to any person or entity with respect to any loss or damage in connection with or arising from the information contained in this reference guide.

Trademarks

Moneris and the Moneris Solutions logo are registered trademarks of Moneris Solutions Corporation.

Any software, hardware and or technology products named in this document are claimed as trademarks or registered trademarks of their respective companies.

Printed in Canada.

10 9 8 7 6 5 4 3 2 1