eSelect PLUS

Merchant Integration Guide

COM API (winHTTP 5.1) – v 1.1.5 (With Status)

Moneris SOLUTIONS

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 credit card transactions. In particular, it describes the format for sending transactions and the corresponding responses you will receive.

2. System and Skill Requirements

In order to use the COM API 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: Your solution may be required to demonstrate compliant with the card associations' PCI/CISP/PABP requirements. For more information please contact our Sales centre and visit http://www.monerisusa.com/pcisecurity

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

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

3. Verified by Visa

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

If you have enrolled in Verified by Visa (VbV) with Moneris and eSELECTplus, please also refer to the *COM* VbV / SecureCode MPI document found at:
https://esplusqa.moneris.com/connect/en/documents/index.html

4. MasterCard SecureCode

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

Before approving a transaction, eSELECTplus and the Bank that issued the MasterCard will authenticate the cardholder through the use of this password. For merchants who have enrolled in SecureCode, please also refer to the *COM* VbV / SecureCode MPI document found at:
https://esplusqa.moneris.com/connect/en/documents/index.html

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

You will need to follow these steps.

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

6. Transaction Types and Transaction Flow

eSELECTplus supports a wide variety of transactions through the API. Below is a list of transactions supported by the API, other terms used for the transaction type are indicated in brackets.

Basic Transactions

Purchase – (sale) The Purchase transaction verifies funds on the customer’s card, removes the funds and readies them for deposit into the merchant’s account.

PreAuth – (authorisation / preauthorisation) The PreAuth verifies and locks funds on the customer’s credit card. The funds are locked for a specified amount of time, based on the card issuer. To retrieve the funds from a PreAuth so that they may be settled in the merchant’s account a Capture must be performed.

Capture – (Completion / PreAuth Completion) Once a PreAuth is obtained the funds that are locked need to be retrieved from the customer’s credit card. The Capture retrieves the locked funds and readies them for settlement into the merchant’s account.

Void – (Correction / Purchase Correction) Purchases and Captures can be voided the same day* that they occur. A Void must be for the full amount of the transaction and will remove any record of it from the cardholder’s statement.

Refund – (Credit) A Refund can be performed against a Purchase or a Capture to refund any part, or all of the transaction.

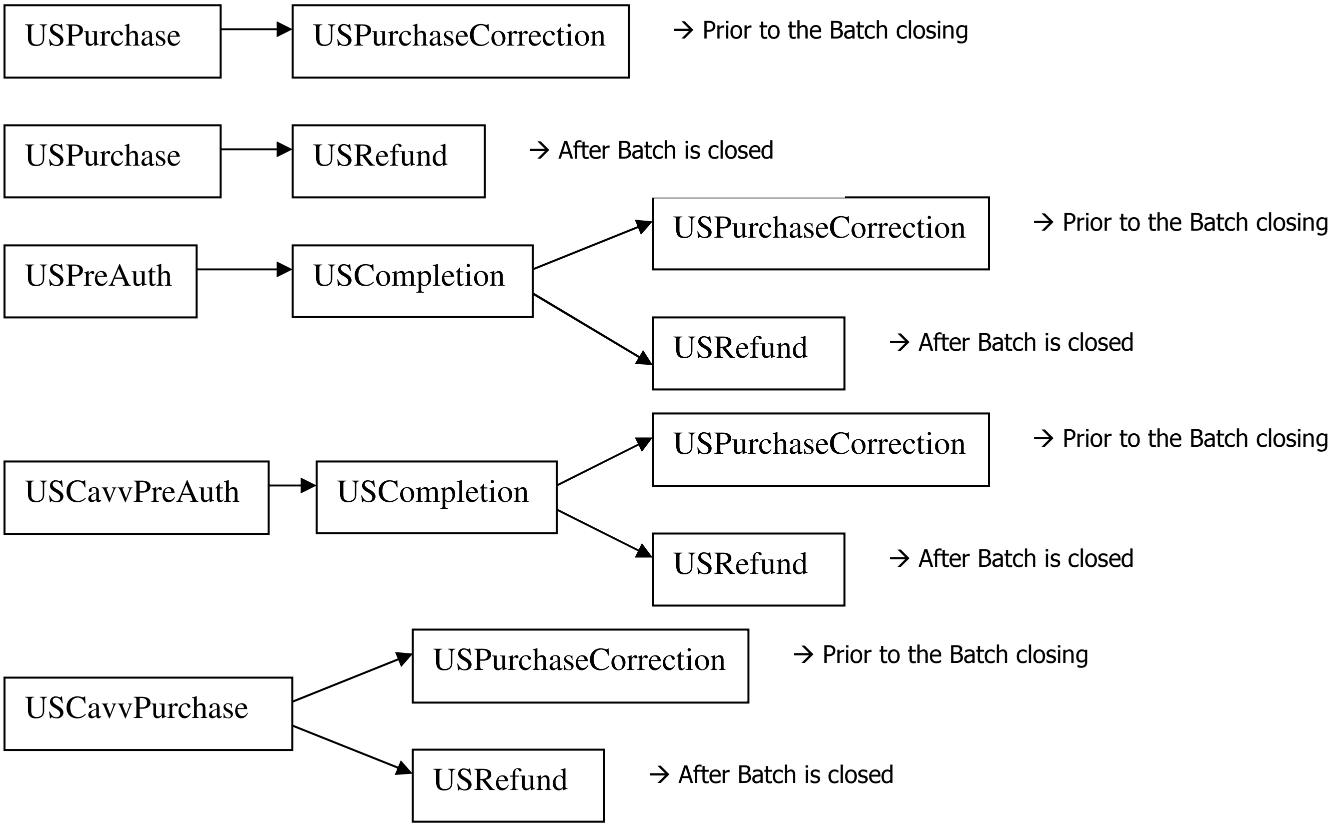
Independent Refund – (Credit) An Independent Refund can be performed to credit money to a Credit Card. This transaction does not require a prior Purchase or Capture.

Batch Close – (End of Day / Settlement) When a Batch Close is performed it takes the monies from all Purchase, Capture and Refund transactions so they will be deposited or debited the following business day. For funds to be deposited the following business day the batch must close before 11pm EST.

Open Totals – (Current Batch Report) When an Open Totals is performed it returns the details about the currently open Batch. This transaction is similar to the Batch Close, though it does not close the Batch for settlement.

* A Void can be performed against a transaction as long as the batch that contains the original transaction remains open.

Process Flow for Basic Credit Card Transactions



Transactions with no Follow-on Required

USIndependentRefund

USBatchClose

USOpenTotals

Mag Swipe Transactions

Mag Swipe Purchase – (sale) The Mag Swipe Purchase transaction requires a credit card to be swiped. It then verifies funds on the customer's card, removes the funds and readies them for deposit into the merchant's account.

Mag Swipe PreAuth – (authorisation / preauthorisation) The Mag Swipe PreAuth requires a credit card to be swiped. It then verifies and locks funds on the customer's credit card. The funds are locked for a specified amount of time, based on the card issuer. To retrieve the funds from a Mag Swipe PreAuth so that they may be settled in the merchant's account a Mag Swipe Capture must be performed.

Mag Swipe Capture – (Completion / PreAuth Completion) Once a Mag Swipe PreAuth is obtained the funds that are locked need to be retrieved from the customer's credit card. The Mag Swipe Capture retrieves the locked funds and readies them for settlement into the merchant's account.

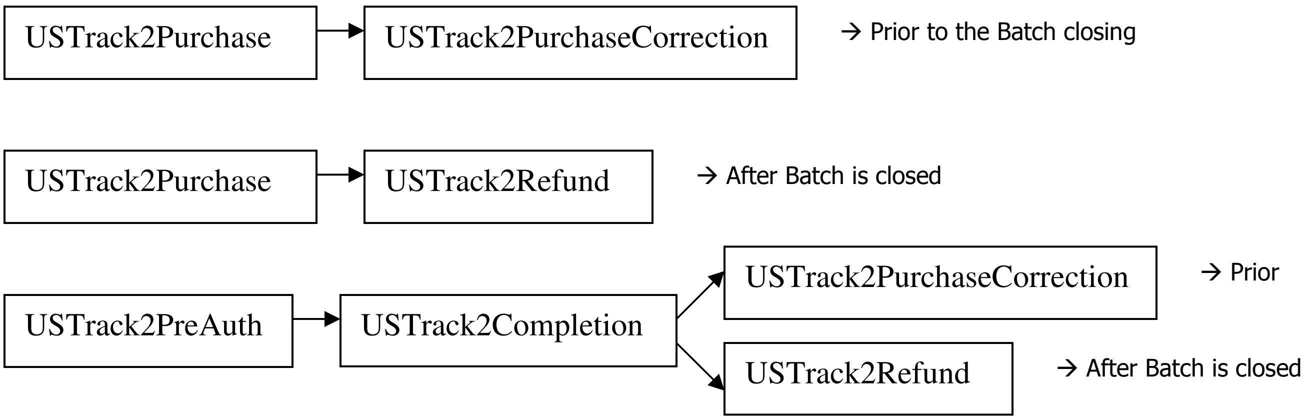
Mag Swipe Void – (Correction / Purchase Correction) Mag Swipe Purchases and Mag Swipe Captures can be voided the same day* that they occur. A Mag Swipe Void must be for the full amount of the transaction and will remove any record of it from the cardholder's statement.

Mag Swipe Refund – (Credit) A Mag Swipe Refund can be performed against a Mag Swipe Purchase or a Mag Swipe Capture to refund any part, or all of the transaction.

Mag Swipe Independent Refund – (Credit) A Mag Swipe Independent Refund requires a credit card to be swiped. It can be performed to credit money to this particular credit card. This transaction does not require a prior Mag Swipe Purchase or Mag Swipe Capture.

* A Void can be performed against a transaction as long as the batch that contains the original transaction remains open.

Process Flow for Mag Swipe Credit Card Transactions



Transactions with no Follow-on Required

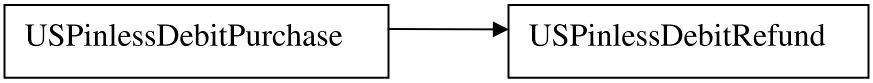
USTrack2IndependentRefund

Pinless Debit Transactions

Pinless Debit Purchase – (sale) A Pinless Debit Purchase transaction verifies funds on the customer's card, removes the funds and readies them for deposit into the merchant's account

Pinless Debit Refund – (Credit) A Pinless Debit Refund transaction can be performed against a Pinless Debit Purchase. No amount is required because the Pinless Debit Refund is always for the full amount of the original transaction.

Process Flow for Pinless Debit Transactions



→ Prior to the Batch closing

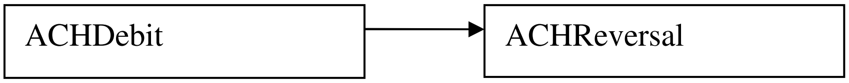
ACH Transactions

ACH Debit – The ACH Debit transaction verifies and collects the customer's bank account information, removes the funds directly from their bank account and readies them for deposit into the merchant's account.

ACH Reversal – The ACH Reversal transaction can be performed against a previously completed ACH Purchase transaction, the full amount of the original ACH Debit transaction will be refunded. An ACH Reversal may only be performed as long as the ACH Debit was performed within the last 3 months.

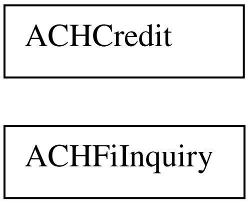
ACH Credit – The ACH Credit transaction verifies and collects the customer's bank account information to allow the merchant to transfer funds from their own bank account directly into the customer's bank.

Process Flow for ACH Transactions



→ Prior or After Batch is closed

Transactions with no Follow-on Required



7. Basic Transaction Examples

Included below is the sample code that can be found in the "Examples" folder of the COM API download.

Purchase (basic)

In the Purchase example we require several variables (store_id, api_token, order_id, amount, pan, expiry_date, and crypt). There are also two optional Level 2 variables (commcard_invoice and commcard_tax_amount) available for Corporate Purchasing Cards. Please refer to Appendix A. Definition of Request Fields for variable definitions.

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

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

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"

customer_id = "Customer_identifier"
commcard_invoice = "Invoice 5757FRJ8"
commcard_tax_amount = "0.15"

Set out = server.CreateObject("Moneris.USRequest")
out.setStatusCheck ("true")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set purchase = server.CreateObject("Moneris.USPurchase")

'Set optional Customer ID & Level 2 details
purchase.setCustId( customer_id )
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount )

out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, crypt_type )
out.sendRequest

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 "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

PreAuth (basic)

The PreAuth is virtually identical to the Purchase with the exception of the transaction type. It is ‘USPreAuth’ instead of ‘USPurchase’. Like the Purchase example, PreAuth’s require several variables (store_id, api_token, order_id, amount, pan, expiry_date, and crypt). Please refer to Appendix A. Definition of Request Fields for variable definitions. In the example below the test environment store_id (monusqa002) and api_token (qatoken) have been hard coded, and highlighted in the HttpsPostRequest object.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "5454545442424242"
exp_date = "0812"
crypt_type = "7"

Set out = server.CreateObject("Moneris.USRequest")
out.setStatusCheck ("true")
out.initRequest "monusqa002", "qatoken", "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set preauth = server.CreateObject("Moneris.USPreauth")

'Set optional Customer ID
preauth.setCustId "My Cust ID"

out.setRequest preauth.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

Capture

The Capture transaction is used to secure the funds locked by a PreAuth transaction. When sending a 'USCompletion' request you will need two pieces of information from the original PreAuth – the order_id and the txn_number from the returned response. There are also two optional Level 2 variables (commcard_invoice and commcard_tax_amount) that may be submitted for Corporate Purchasing Cards.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%

store_id = "monusqa002"
api_token = "qatoken"

order_id = "mjr-1112006-1696"
comp_amount = "1.00"
txn_number = "1089-0_1"
crypt_type = "7"

commcard_invoice = "Invoice 5757FRJ8"
commcard_tax_amount = "0.15"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set capture = server.CreateObject("Moneris.USCompletion")

'set optional Level 2 details
capture.setCommcardInvoice( commcard_invoice )
capture.setCommcardTaxAmount( commcard_tax_amount )

out.setRequest capture.formatRequest( order_id, comp_amount, txn_number, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

Void

The Void (USPurchaseCorrection) transaction is used to cancel a transaction that was performed in the current batch. No amount is required because a Void is always for 100% of the original transaction. The only transactions that can be Voided are Captures and Purchases. To send a 'USPurchaseCorrection' the order_id and txn_number from the 'USCompletion' or 'USPurchase' are required.

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

order_id = "mjr-1112006-162358"
txn_number = "1093-0_1"
crypt_type = "7"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set void = server.CreateObject("Moneris.USPurchaseCorrection")
out.setRequest void.formatRequest( order_id, txn_number, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>
</BODY>
</HTML>
```

Refund

The Refund will credit a specified amount to the cardholder's credit card. A Refund can be sent up to the full value of the original Capture or Purchase. To send a 'USRefund' you will require the order_id and txn_number from the original 'USCompletion' or 'USPurchase'.

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

order_id = "mjr-1112006-16059"
amount = "1.00"
txn_number = "1087-0_1"
crypt_type = "7"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set refund = server.CreateObject("Moneris.USRefund")
out.setRequest refund.formatRequest( order_id, amount, txn_number, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

Independent Refund

The Independent Refund (USIndependentRefund) will credit a specified amount to the cardholder's credit card. The Independent Refund does not require an existing order to be logged in the eSELECTplus gateway; however, the credit card number and expiry date will need to be passed. In the example below the values are hard coded (and highlighted) in the appropriate locations. The transaction format is almost identical to a Purchase or a PreAuth.

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

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set indRefund = server.CreateObject("Moneris.USIndependentRefund")

'Set Optional Customer ID
indRefund.setCustId "My Cust ID"

out.setRequest indRefund.formatRequest("unique_order_id", "1.00", "4242424242424242", "0912", "7")
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

Force Post

The Force Post (USForcepost) transaction is used when a merchant obtains the authorization number directly from the issuer using a phone or any third party authorization method. The Force Post does not require an existing order to be logged in the eSELECTplus gateway; however, the credit card number, expiry date and the authorization number will need to be passed. In the example below the values are hard coded in the appropriate locations.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
order_id = "SL-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
auth_code = "123456"
customer_id = "Customer_identifier"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set forcePost = server.CreateObject("Moneris.USForcepost")

'Set optional Customer ID & Level 2 details
forcePost.setCustId( customer_id )

out.setRequest forcePost.formatRequest( order_id, amount, pan, exp_date, auth_code, 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 "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

8. Basic Transactions with Extra Features - Examples

In the previous section the instructions were provided for the basic transaction set. eSELECTplus also provides several extra features/functionalities for the basic transactions. These features include storing customer and order details, Verified by Visa / SecureCode and sending transactions to the Recurring Billing feature. Verified by Visa / SecureCode and Recurring Billing must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated.

Purchase (with Customer and Order details)

Below is an example of sending a Purchase with the customer and order details. If one piece of information 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. The identical format is used for PreAuth with the exception of transaction type which changes from 'USPurchase' to 'USPreAuth'. Customer details can only be sent with Purchase and PreAuth. It can be used in conjunction with other extra features such as VBV/MCSC and Recurring Billing. ***Please note that the CustInfo fields are not used for any type of address verification or fraud check.***

```
<html>
<head>
       <title>Untitled</title>
</head>
<body>

<%
store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject( "Moneris.USPurchase" )

'=======================================
'Setting optional Customer ID
purchase.setCustId("My Cust ID")

'=======================================
'Setting Corporate Level 2 data
purchase.setCommcardInvoice("Invoice abs4567")
purchase.setCommcardTaxAmount("0.17")

'=======================================
'Setting Customer Info data.
first_name = "MyFirst_Name"
last_name = "MyLast_Name"
company_name = "My Company name"
address = "My Address"
city = "MyCity"
province = "MyProvince"
postal_code = "A3A B5B"
country = "MyCountry"
phone_number = "416-333-5555"
fax = "416-333-6666"
tax1 = "0.07"
tax2 = "0.08"
tax3 = "1.00"
shipping_cost = "6.50"
email = "q@q.com"
instruction = "Make it so!"
ShippingAddr = purchase.formatAddress( first_name, last_name, company_name, address, city, province,
postal_code, country, phone_number, fax, tax1, tax2, tax3, shipping_cost )
BillingAddr = ShippingAddr

item_name = "Hammer"
item_quantity = "1"
product_code = "hmr-001"
item_amount = "6.99"
```

```
purchase.setItem item_name, item_quantity, product_code, item_amount

purchase.setCustInfo BillingAddr, ShippingAddr, email, instruction


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, crypt_type )
out.sendRequest


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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"

%>
</body>
</html>
```

Purchase (with Verified by Visa / MasterCard SecureCode)

Below is an example of sending a Purchase with the Verified by Visa / SecureCode extra fields. The ‘cavv’ is obtained by using either the Moneris MPI or a third party MPI. The format outlined below is identical for a PreAuth with the exception of the TransType which changes from ‘USCavvPurchase’ to ‘USCavvPreAuth’. VBV/MCSC must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated. The optional customer and order details can be included in the transaction using the method outlined above -*Purchase (with Customer and Order Details).*

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
        store_id = “monusqa006”
        api_token = “qatoken”
        order_id = “need_original_from_MPI”
        cavv = “need_value_from_MPI”

        Set out = Server.CreateObject( "Moneris.USRequest" )
        out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

        Set purreq = Server.CreateObject( "Moneris.USCAVVPurchase" )
        out.setRequest purreq.formatRequest( order_id, amount, pan, exp_date, cavv )

        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 "Bank Totals: " & out.getBankTotals & "<br>"
        Response.Write "Card Type: " & out.getCardType & "<br>"
        Response.Write "Reference Number: " & out.getReferenceNum & "<br>"
        Response.Write "Transaction ID: " & out.getTransID & "<br>"
        Response.Write "Auth Code: " & out.getAuthCode & "<br>"
        Response.Write "Transaction Time: " & out.getTransTime & "<br>"
        Response.Write "Transaction Date: " & out.getTransDate & "<br>"
        Response.Write "Complete: " & out.getCompleteStatus & "<br>"
        Response.Write "Timeout: " & out.getTimedoutStatus & "<br>"
        Response.Write "Ticket: " & out.getTicket & "<br>"
%>
</BODY>
</HTML>
```

Purchase (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 and Recur Update 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 -*Purchase (with Customer and Order Details).* 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"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USPurchase")

'=======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'=======================================
'Set optional Level 2 details
commcard_invoice = "Invoice 56789"
commcard_tax_amount = "0.08"
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount)

'=======================================
'Recurring setup.
recur_unit = "day"
start_now = "true"
start_date = "2006/12/14"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
purchase.setRecur recur_unit, start_now, start_date, num_recurs, period, recur_amount


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "RecurSuccess:  " & out.getRecurSuccess & "<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.

Purchase (with CVD and AVS - eFraud)

Below is an example of a Purchase transaction with CVD and AVS information. These values can be sent in conjunction with other additional variables such as Recurring Billing or customer information. With this feature enabled in your merchant profile, you will be able to pass in these fields for the following transactions 'USPurchase', 'USPreAuth', 'USCavvPurchase', and 'USCavvPreAuth'. To form CvdInfo please refer to Appendix K. Card Validation Digits (CVD), to form AvsInfo please refer to Appendix L. Address Verification Service (AVS).

If you wish to have the eFraud feature added to your profile, once you have completed your testing, please contact the eSELECTplus Integration Support Team at eselectplus@moneris.com to have it enabled. Please make sure to include your store ID or merchant number in any eFraud activation email requests.

When testing eFraud (AVS and CVD) 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://esplusqa.moneris.com/connect/en/documents/index.html

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"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USPurchase")

'======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'======================================
'Set Optional Level 2 Details
purchase.setCommcardInvoice("Invoice 123456")
purchase.setCommcardTaxAmount("0.07")


'======================================
'EFraud setup.
avs_street_number = "123"
avs_street_name = "East Street"
avs_zipcode = "M1M2M2"
purchase.setAvsInfo avs_street_number, avs_street_name, avs_zipcode

cvd_indicator = "1"
cvd_value = "123"
purchase.setCvdInfo cvd_indicator, cvd_value


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "10.42"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "AVS Result Code:  " & out.getAvsResultCode & "<br>"
Response.Write "CVD Result Code:  " & out.getCvdResultCode & "<br>"
%>

</BODY>
</HTML>
```

As part of the eFraud response there will be two additional methods called getAvsResultCode( ) and getCvdResultCode( ). For a list of possible CVD responses please refer to Appendix K. Card Validation Digits (CVD)f and for a list of AVS responses, please refer to Appendix L. Address Verification Service (AVS).

9. Mag Swipe Transaction Examples

Included below is the sample code for the Mag Swipe transactions that can be found in the "Examples" folder of the COM API download. Mag Swipe transactions allow the user to swipe their credit card and submit the Track2 details. These transactions support the submission of 'track2', as well as a manual entry of the credit card number and expiry date using the 'pan' and 'expiry_date' variables. If all three fields are submitted, the track2 details will be used to process the transaction.

Mag Swipe Purchase

Similar to the basic Purchase, in the Mag Swipe Purchase (USTrack2Purchase) example we require several variables (store_id, api_token, order_id, amount, track2 and/or pan, expiry_date, and pos_code). There are also two optional Level 2 variables (commcard_invoice and commcard_tax_amount) available for Corporate Purchasing Cards. Please refer to Appendix A. Definition of Request Fields for variable definitions.

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

<script type="text/javascript">

var track

if ((location.href).indexOf("?") == -1)
{
    var track1 = prompt("Please swipe your card", "Please Swipe")

    if (track1.indexOf(";") == 0)
    {
        track = track1
    }
    else
    {
        var track2 = prompt("Please swipe your card again", "Please Swipe to get track2")
        track = track2
    }
    document.write("Track2 = " + track)
}

function setTrack2 (form)
{
       form.track2.value=track
}

</script>

<form name="paymentForm" method="get">
    Is this track2 correct? <input type="checkbox" onClick="setTrack2(this.form)"/><br/>
       <input type="hidden" name="track2" value=""/>
    <input type="submit" value="Submit" />
</form>

<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
store_id = "monusqa002"
api_token = "qatoken"

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
track2 = Request("track2")
pan = ""
exp_date = ""
pos_code = "00"
```

```
customer_id = "Customer_identifier"
commcard_invoice = "Invoice 5757FRJ8"
commcard_tax_amount = "0.15"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set purchase = server.CreateObject("Moneris.USTrack2Purchase")

'Set optional Customer ID & Level 2 details
purchase.setCustId( customer_id )
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount )

out.setRequest purchase.formatRequest( order_id, amount, track2, pan, exp_date, pos_code )
out.sendRequest

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 "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "StatusCode:  " & out.getStatusCode & "<br>"
Response.Write "StatusMsg:  " & out.getStatusMsg & "<br>"

%>

</BODY>
</HTML>
```

Mag Swipe PreAuth

The Mag Swipe PreAuth is virtually identical to the Purchase with the exception of the transaction type. It is 'USTrack2PreAuth' instead of 'USTrack2Purchase'. Like the Purchase example, PreAuth's require several variables (store_id, api_token, order_id, amount, track2 and/or pan, expiry_date, and pos_code). Please refer to Appendix A. Definition of Request Fields for variable definitions. In the example below the test environment store_id (monusqa002) and api_token (qatoken) have been hard coded, and highlighted in the HttpsPostRequest object.

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

<script type="text/javascript">

var track

if ((location.href).indexOf("?") == -1)
{
    var track1 = prompt("Please swipe your card", "Please Swipe")

    if (track1.indexOf(";") == 0)
    {
        track = track1
    }
    else
    {
        var track2 = prompt("Please swipe your card again", "Please Swipe to get track2")
        track = track2
    }
    document.write("Track2 = " + track)
}

function setTrack2 (form)
{
    form.track2.value=track
}

</script>

<form name="paymentForm" method="get">
    Is this track2 correct? <input type="checkbox" onClick="setTrack2(this.form)"/><br/>
       <input type="hidden" name="track2" value=""/>
    <input type="submit" value="Submit" />
</form>

<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
track2 = Request("track2")
pan = ""
exp_date = ""
pos_code = "00"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest "monusqa002", "qatoken", "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set preauth = server.CreateObject("Moneris.USTrack2Preauth")

'Set optional Customer ID
preauth.setCustId "My Cust ID"

out.setRequest preauth.formatRequest( order_id, amount, track2, pan, exp_date, pos_code )
out.sendRequest
```

```
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Mag Swipe Capture

The Mag Swipe Capture (USTrack2Completion) transaction is used to secure the funds locked by a 'USTrack2PreAuth' transaction. When sending a 'USTrack2Completion' request you will need two pieces of information from the original 'USTrack2PreAuth' – the order_id and the txn_number from the returned response; it does not require the customer to re-swipe the credit card. There are also two optional Level 2 variables (commcard_invoice and commcard_tax_amount) that may be submitted for Corporate Purchasing Cards.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
store_id = "monusqa002"
api_token = "qatoken"

order_id = "mjr-3112006-132150"
comp_amount = "1.00"
txn_number = "1203-0_1"
pos_code = "00"

commcard_invoice = "Invoice 5757FRJ8"
commcard_tax_amount = "0.15"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set capture = server.CreateObject("Moneris.USTrack2Completion")

'set optional Level 2 details
capture.setCommcardInvoice( commcard_invoice )
capture.setCommcardTaxAmount( commcard_tax_amount )

out.setRequest capture.formatRequest( order_id, comp_amount, txn_number, pos_code )
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Mag Swipe Void

The Mag Swipe Void (USTrack2PurchaseCorrection) transaction is used to cancel a transaction that was performed in the current batch. No amount is required because a Void is always for 100% of the original transaction. The only transactions that can be Voided are Captures and Purchases. To send a 'USTrack2PurchaseCorrection' the order_id and txn_number from the 'USTrack2Completion' or 'USTrack2Purchase' are required; it does not require the customer to re-swipe the credit card.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
store_id = "monusqa002"
api_token = "qatoken"

order_id = "mjr-3112006-132150"
txn_number = "1204-1_1"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set void = server.CreateObject("Moneris.USTrack2PurchaseCorrection")
out.setRequest void.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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>
</BODY>
</HTML>
```

Mag Swipe Refund

The Mag Swipe Refund (USTrack2Refund) will credit a specified amount to the cardholder's credit card. A Refund can be sent up to the full value of the original Capture or Purchase. To send a 'USTrack2Refund' you will require the order_id and txn_number from the original 'USTrack2Completion' or 'USTrack2Purchase'; it does not require the customer to re-swipe the credit card.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
store_id = "monusqa002"
api_token = "qatoken"

order_id = "mjr-3112006-13119"
amount = "1.00"
txn_number = "1201-0_1"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set refund = server.CreateObject("Moneris.USTrack2Refund")
out.setRequest refund.formatRequest( order_id, amount, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Mag Swipe Independent Refund

The Mag Swipe Independent Refund (USTrack2IndependentRefund) will credit a specified amount to the cardholder's credit card. The Mag Swipe Independent Refund does not require an existing order to be logged in the eSELECTplus gateway; however, the credit card will need to be swiped to provide the track2. In the example below the values are hard coded in the appropriate locations. The transaction format is almost identical to a Mag Swipe Purchase or a Mag Swipe PreAuth.

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

<script type="text/javascript">

var track

if ((location.href).indexOf("?") == -1)
{
    var track1 = prompt("Please swipe your card", "Please Swipe")

    if (track1.indexOf(";") == 0)
    {
        track = track1
    }
    else
    {
        var track2 = prompt("Please swipe your card again", "Please Swipe to get track2")
        track = track2
    }
    document.write("Track2 = " + track)
}

function setTrack2 (form)
{
       form.track2.value=track
}

</script>

<form name="paymentForm" method="get">
    Is this track2 correct? <input type="checkbox" onClick="setTrack2(this.form)"/><br/>
       <input type="hidden" name="track2" value=""/>
    <input type="submit" value="Submit" />
</form>

<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest "monusqa002", "qatoken", "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set indRefund = server.CreateObject("Moneris.USTrack2IndependentRefund")

'Set Optional Customer ID
indRefund.setCustId "My Cust ID"

out.setRequest indRefund.formatRequest("unique_order_id", "1.00", Request("track2"), "", "", "00" )
out.sendRequest
```

```
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Mag Swipe Forcepost

The Mag Swipe Force Post (USTrack2ForcePost) is used when a merchant obtains the authorization number directly from the issuer using a phone or any third party authorization method. The Mag Swipe Force Post does not require an existing order to be logged in the eSELECTplus gateway; however, the credit card will need to be swiped to provide the track2 data. To complete the transaction, the authorization number obtained from the issuer must also be entered. In the example below the values are hard coded in the appropriate locations.

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

<script type="text/javascript">

var track

if ((location.href).indexOf("?") == -1)
{
    var track1 = prompt("Please swipe your card", "Please Swipe")

    if (track1.indexOf(";") == 0)
    {
        track = track1
    }
    else
    {
        var track2 = prompt("Please swipe your card again", "Please Swipe to get track2")
        track = track2
    }
    document.write("Track2 = " + track)
}

function setTrack2 (form)
{
       form.track2.value=track
}

</script>

<form name="paymentForm" method="get">
    Is this track2 correct? <input type="checkbox" onClick="setTrack2(this.form)"/><br/>
       <input type="hidden" name="track2" value=""/>
    <input type="submit" value="Submit" />
</form>

<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
track2 = Request("track2")
Response.Write ("Card Number: " & track2 & "<br>")
pan = ""
exp_date = ""
pos_code = "00"
auth_code = "123456"
customer_id = "Customer_identifier"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set forcepost = server.CreateObject("Moneris.USTrack2Forcepost")

'Set optional Customer ID & Level 2 details
forcepost.setCustId( customer_id )

out.setRequest forcepost.formatRequest( order_id, amount, track2, pan, exp_date, auth_code, pos_code )
out.sendRequest

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 "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

10. Mag Swipe Transactions with Extra Features - Examples

In the previous section the instructions were provided for the Mag Swipe transaction set. eSELECTplus also provides the ability to perform an Address Verification Service check with a Mag Swipe transaction.

Mag Swipe Purchase (with Address Verification Service – AVS)

Below is an example of a Mag Swipe Purchase transaction with AVS information. With this feature enabled in your merchant profile, you will be able to pass in these fields for both 'USTrack2Purchase' and 'USTrack2PreAuth' transactions. To form AvsInfo please refer to Appendix L. Address Verification Service (AVS).

If you wish to have the AVS (eFraud) feature added to your profile, once you have completed your testing, please contact the eSELECTplus Integration Support Team at eselectplus@moneris.com to have it enabled. Please make sure to include your store ID or merchant number in any eFraud activation email requests.

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://esplusqa.moneris.com/connect/en/documents/index.html

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

<script type="text/javascript">

var track

if ((location.href).indexOf("?") == -1)
{
    var track1 = prompt("Please swipe your card", "Please Swipe")

    if (track1.indexOf(";") == 0)
    {
        track = track1
    }
    else
    {
        var track2 = prompt("Please swipe your card again", "Please Swipe to get track2")
        track = track2
    }
    document.write("Track2 = " + track)
}

function setTrack2 (form)
{
    form.track2.value=track
}

</script>

<form name="paymentForm" method="get">
    Is this track2 correct? <input type="checkbox" onClick="setTrack2(this.form)"/><br/>
       <input type="hidden" name="track2" value=""/>
    <input type="submit" value="Submit" />
</form>

<% @Language = "VBScript" %>
<% Response.buffer = true %>

<%
store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
```

```
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USTrack2Purchase")

'=======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'=======================================
'Set Optional Level 2 Details
purchase.setCommcardInvoice("Invoice 123456")
purchase.setCommcardTaxAmount("0.07")

'=======================================
'EFraud - AVS setup.
avs_street_number = "123"
avs_street_name = "East Street"
avs_zipcode = "M1M2M2"
purchase.setAvsInfo avs_street_number, avs_street_name, avs_zipcode

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "10.13"
track2 = Request("track2")
Response.Write ("Card Number: " & track2 & "<br>")
pan = "4242424242424242"
exp_date = "0909"
pos_code = "00"
out.setRequest purchase.formatRequest( order_id, amount, track2, pan, exp_date, pos_code )
out.sendRequest

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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "AVS Result Code:  " & out.getAvsResultCode & "<br>"
%>

</BODY>
</HTML>
```

As part of the AVS (eFraud) response there will be an additional method called getAvsResultCode( ). For a list of possible AVS responses, please refer to Appendix L. Address Verification Service (AVS)

NOTE

Please note, the above transaction with AVS information may not be tested within our current test environment because the test Visa card numbers can not be swiped. To receive an AVS response you may pass the test Visa card number as the 'pan'.

11. Pinless Debit Transaction Examples

Included below is the sample code for the Pinless Debit transactions that can be found in the “Examples” folder of the COM API download. Pinless Debit transactions allow the user to submit billing invoice account information and have funds debited from their bank account for bill payment. This transaction type lets the merchant know if the funds are available or not.

Pinless Debit Purchase

In the Pinless Debit Purchase (USPinlessDebitPurchase) example we require several mandatory variables: store_id, api_token, order_id, amount, pan, presentation_type, intended_use and p_account_number. There are also many optional variables, such as the expiry_date, cust_id. Please refer to Appendix A. Definition of Request Fields and Appendix E. Pinless Debit Fields for variable definitions.

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

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

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4496270000164824"
presentation_type = "W"
intended_use = "1"
p_account_number = "1234567890123456789012345"

customer_id = "my cust id"
expdate = "0909"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set pinlessDebitPurchase = server.CreateObject("Moneris.USPinlessDebitPurchase")

'Set optional variables
pinlessDebitPurchase.setCustId( customer_id )
pinlessDebitPurchase.setExpDate( expdate )

out.setRequest pinlessDebitPurchase.formatRequest( order_id, amount, pan, presentation_type, intended_use,
p_account_number )
out.sendRequest

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 "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Pinless Debit Refund

The Pinless Debit Refund (USPinlessDebitRefund) transaction is used to refund a prior Pinless Debit Purchase transaction that was performed within the past 3 months. No amount is required because a Pinless Debit Refund is always for 100% of the original transaction. To send a 'USPinlessDebitRefund' the order_id and txn_number from the 'USPinlessDebitPurchase' are required; it does not require the Pinless Debit information to be re-entered.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
order_id = "mjr-1112006-16059"
txn_number = "1087-0_1"
amount = "1.00"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set pinlessDebitRefund = server.CreateObject("Moneris.USPinlessDebitRefund")

'Set optional variable
pinlessDebitRefund.setAmount( amount )

out.setRequest pinlessDebitRefund.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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

12. Pinless Debit Transaction with Extra Features - Examples

Pinless Debit Purchase (with Customer and Order details)

In the Pinless Debit Purchase (USPinlessDebitPurchase) example we require several mandatory variables: store_id, api_token, order_id, amount, pan, presentation_type, intended_use and p_account_number. There is also an optional variable, such as the expiry_date, cust_id. Please refer to Appendix A. Definition of Request Fields, Appendix C. CustInfo Fields and Appendix E. Pinless Debit Fields for variable definitions.

```
<html>
<head>
       <title>Untitled</title>
</head>
<body>
<%

store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host


Set pinlessDebitPurchase = server.CreateObject("Moneris.USPinlessDebitPurchase")

'=======================================
'Setting optional variables
pinlessDebitPurchase.setCustId( "customer id" )
pinlessDebitPurchase.setExpDate( "0909" )

'=======================================
'Setting Customer Info data.
first_name = "MyFirst_Name"
last_name = "MyLast_Name"
company_name = "My Company name"
address = "My Address"
city = "MyCity"
province = "MyProvince"
postal_code = "A3A B5B"
country = "MyCountry"
phone_number = "416-333-5555"
fax = "416-333-6666"
tax1 = "0.07"
tax2 = "0.08"
tax3 = "1.00"
shipping_cost = "6.50"
email = "q@q.com"
instruction = "Make it so!"
ShippingAddr = pinlessDebitPurchase.formatAddress( first_name, last_name, company_name, address, city,
province, postal_code, country, phone_number, fax, tax1, tax2, tax3, shipping_cost )
BillingAddr = ShippingAddr

item_name = "Hammer"
item_quantity = "1"
product_code = "hmr-001"
item_amount = "6.99"
pinlessDebitPurchase.setItem item_name, item_quantity, product_code, item_amount

pinlessDebitPurchase.setCustInfo BillingAddr, ShippingAddr, email, instruction


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4496270000164824"
presentation_type = "W"
intended_use = "1"
p_account_number = "1234567890123456789012345"
out.setRequest pinlessDebitPurchase.formatRequest( order_id, amount, pan, presentation_type, intended_use,
p_account_number )
out.sendRequest
```

```
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"

%>
</body>
</html>
```

Pinless Debit Purchase (with Recurring Billing)

The Pinless Debit Purchase with Recurring Billing (USPinlessDebitPurchase) transaction allows the merchant to submit the transaction information once and then re-bill 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. Please see Appendix A. Definition of Request Fields, Appendix D. Recur and Recur Update Fields and Appendix E. Pinless Debit Fields for description of each of the fields.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%

store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host


Set pinlessDebitPurchase = server.CreateObject("Moneris.USPinlessDebitPurchase")

'======================================
'Set optional variables
cust_id = "Customer_identifier"
pinlessDebitPurchase.setCustId( cust_id )

expdate = "0909"
pinlessDebitPurchase.setExpDate( expdate )

'======================================
'Recurring setup.
recur_type = "day"
start_now = "true"
start_date = "2008/12/14"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
pinlessDebitPurchase.setRecur recur_type, start_now, start_date, num_recurs, period, recur_amount


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4496270000164824"
presentation_type = "W"
intended_use = "1"
p_account_number = "1234567890123456789012345"
out.setRequest pinlessDebitPurchase.formatRequest( order_id, amount, pan, presentation_type, intended_use,
p_account_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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "RecurSuccess:  " & out.getRecurSuccess & "<br>"
%>

</BODY>
</HTML>
```

13. ACH Transaction Examples

Included below is the sample code for the ACH transactions that can be found in the "Examples" folder of the COM API download. ACH transactions allow the user to submit bank account information to have funds either debited or credited.

ACH Debit

In the ACH Debit (USAchDebit) example we require several mandatory variables: store_id, api_token, order_id, amount, sec, routing_num, account_num, and account_type. There are also many optional variables, such as the cust_id, check_num and the customer details. Please refer to Appendix A. Definition of Request Fields for all request variables and Appendix F. AchInfo Fields for all ACH variables. ***Please note that the AchInfo fields are not used for any type of address verification or fraud check.***

ACH Debit (Check not present)

SEC codes for physical check **not present** include: 'web', 'ccd', 'ppd'. In the example below the following variables are required for an ACH Debit (check not present) transaction: routing_num, account_num, check_num, account_type and micr. Please refer to

Appendix G. ACH Sec Codes and Process Flow for a full description on the mandatory fields.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa024"
api_token = "qatoken"
order_id = "MR-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"

customer_id = "Customer_identifier"

Set out = server.CreateObject("Moneris.USRequest")
'out.setStatusCheck ("true")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achDebit = server.CreateObject("Moneris.USAchDebit")

'Ach Info fields
'The values can be a string, "" or nullvalue

sec = "ppd"
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 = "54321"
account_num = "23456"
check_num = "100"
account_type = "savings"

achDebit.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

'Set optional Customer ID
achDebit.setCustId( customer_id )

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

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<table cellpadding="5" cellspacing="0">
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
        <tr><td>StatusCode:</td><td><%=out.getStatusCode%></td></tr>
        <tr><td>StatusMsg:</td><td><%=out.getStatusMsg%></td></tr>
</table>

</BODY>
</HTML>
```

ACH Credit

In the ACH Credit (ACHCredit) example we require several mandatory variables: store_id, api_token, order_id, amount, sec, routing_num, account_num, and account_type. There are also many optional variables, such as the cust_id, check_num and the customer details. Please refer to Appendix A. Definition of Request Fields and Appendix F. AchInfo Fields for variable definitions. ***Please note that the AchInfo fields are not used for any type of address verification or fraud check.***

NOTE Please note, the ACH Credit transaction may only be submitted with a SEC Code of ‘ppd’ or ‘ccd’.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa024"
api_token = "qatoken"
order_id = "SL-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achCredit = server.CreateObject("Moneris.USAchCredit")

'Ach Info fields
'The values can be a string, "" or nullvalue

sec = "ppd"
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 = "54321"
account_num = "23456"
check_num = "100"
account_type = "savings"

achCredit.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

'Set optional Customer ID
achCredit.setCustId "My Cust ID"

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

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

<table cellpadding="5" cellspacing="0">
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
</table>
</BODY>
</HTML>
```

ACH Reversal

The ACH Reversal (USAchReversal) transaction is used to reverse a prior ACH Debit transaction that was performed within the past 3 months. No amount is required because a Reversal is always for 100% of the original transaction. To send a 'USAchReversal' the order_id and txn_number from the 'ACHDebit' are required; it does not require the bank account information to be re-entered.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa024"
api_token = "qatoken"
order_id = "SL-182007-11220"
txn_number ="162-0_5"
amount = "1.00"


Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achReversal = server.CreateObject("Moneris.USAchReversal")

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

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

<table cellpadding="5" cellspacing="0">
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
</table>


</BODY>
</HTML>
```

ACH FI Enquiry

The ACH FI Enquiry (USAchFiEnquiry) transaction allows the merchant to submit a routing number and verify which Financial Institution it belongs to. This transaction also allows the merchant to verify whether or not this is a valid routing number before submitting an ACH Debit or Credit transaction.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa024"
api_token = "qatoken"
routing_num = "123456"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achFiEnquiry = server.CreateObject("Moneris.USAchFiEnquiry")

out.setRequest achFiEnquiry.formatRequest (routing_num)
out.sendRequest
%>

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

<table cellpadding="5" cellspacing="0">
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
</table>

</BODY>
</HTML>
```

14. ACH Transactions with Extra Features – Examples

In the previous section the instructions were provided for the ACH transaction set. eSELECTplus also provides several extra features/functionalities for the ACH transactions. These features include storing customer and order details and sending transactions to the Recurring Billing feature. Recurring Billing must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated.

ACH Debit (with Customer and Order details)

Below is an example of sending an ACH Debit (USAchDebit) with the customer and order details. If one piece of information 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. Customer details can only be sent with the ACH Debit transaction. It can be used in conjunction with other extra features such as 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 = "monusqa024"
api_token = "qatoken"
order_id = "SL-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"

customer_id = "Customer_identifier"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achDebit = server.CreateObject("Moneris.USAchDebit")

'Ach Info fields
'The values can be a string, "" or nullvalue

sec = "ppd"
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 = "54321"
account_num = "23456"
check_num = "100"
account_type = "savings"

achDebit.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

'Setting Customer Info data.
first_name = "MyFirst_Name"
last_name = "MyLast_Name"
company_name = "My Company name"
address = "My Address"
city = "MyCity"
province = "MyProvince"
postal_code = "A3A B5B"
country = "MyCountry"
phone_number = "416-333-5555"
fax = "416-333-6666"
tax1 = "0.07"
tax2 = "0.08"
tax3 = "1.00"
shipping_cost = "6.50"
email = "q@q.com"
instruction = "Make it so!"
ShippingAddr = achDebit.formatAddress( first_name, last_name, company_name, address, city, province,
postal_code, country, phone_number, fax, tax1, tax2, tax3, shipping_cost )
```

```
BillingAddr = ShippingAddr

item_name = "Hammer"
item_quantity = "1"
product_code = "hmr-001"
item_amount = "6.99"
achDebit.setItem item_name, item_quantity, product_code, item_amount

achDebit.setCustInfo BillingAddr, ShippingAddr, email, instruction

'Set optional Customer ID
achDebit.setCustId( customer_id )

out.setRequest achDebit.formatRequest( order_id, amount )
out.sendRequest
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" Content="text-html; charset=Windows-1252">
</HEAD>
<BODY bgcolor=white>
<table cellpadding="5" cellspacing="0">
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
</table>

</BODY>
</HTML>
```

ACH Debit (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 and Recur Update Fieldsfor descry iption of each of the Recur fields. The optional customer and order details can be included in the transaction using the method outlined above – *ACH Debit (with Customer and Order Details)*. Recurring Billing must be added to your account, please call the Service Centre at 1-866-423-8475 to have your profile updated. ***Please note that the Recurring Billing fields are only available to SEC codes 'ppd' 'ccd' and 'web'.***

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa024"
api_token = "qatoken"
order_id = "SL-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"

customer_id = "Customer_identifier"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id , api_token , "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set achDebit = server.CreateObject("Moneris.USAchDebit")


'Ach Info fields
'The values can be a string, "" or nullvalue

sec = "ppd"
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 = "54321"
account_num = "23456"
check_num = "100"
account_type = "savings"

achDebit.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

'Recurring setup.
recur_type = "day"
start_now = "true"
start_date = "2007/08/02"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
achDebit.setRecur recur_type, start_now, start_date, num_recurs, period, recur_amount

'Set optional Customer ID
achDebit.setCustId( customer_id )

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

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

<table cellpadding="5" cellspacing="0">
```

```
        <tr><td>Receipt ID:</td><td><%=out.getReceiptID%></td></tr>
        <tr><td>Response Code:</td><td><%=out.getResponseCode%></td></tr>
        <tr><td>Transaction Type:</td><td><%=out.getTransType%></td></tr>
        <tr><td>Message:</td><td><%=out.getMessage%></td></tr>
        <tr><td>Amount:</td><td><%=out.getTransAmount%></td></tr>
        <tr><td>Card Type:</td><td><%=out.getCardType%></td></tr>
        <tr><td>Reference Number:</td><td><%=out.getReferenceNum%></td></tr>
        <tr><td>Transaction ID:</td><td><%=out.getTransID%></td></tr>
        <tr><td>Auth Code:</td><td><%=out.getAuthCode%></td></tr>
        <tr><td>Transaction Time:</td><td><%=out.getTransTime%></td></tr>
        <tr><td>Transaction Date:</td><td><%=out.getTransDate%></td></tr>
        <tr><td>Complete:</td><td><%=out.getCompleteStatus%></td></tr>
        <tr><td>Timeout:</td><td><%=out.getTimedoutStatus%></td></tr>
        <tr><td>Ticket:</td><td><%=out.getTicket%></td></tr>
</table>



</BODY>
</HTML>
```

15. Administrative Transactions

Included below is the sample code for the Administrative transactions that can be found in the “Examples” folder of the COM API download. Administrative transactions allow the user to perform such tasks as manually closing an open Batch and preparing the funds for settlement. Also, the user may retrieve details about the currently open Batch without needing to close it.

Batch Close

At the end of every financial day (11pm EST) the Batch needs to be closed in order to have the Credit Card funds settled the next business day and the ACH funds settled, on average, within the next 5 business days. *By default eSELECTplus will automatically close your Batch daily, whenever there are funds in the open Batch.* Some merchants prefer to control Batch Close, and disable the automatic functionality. For these merchants we have provided the ability to close your Batch through the API. When a Batch is closed the response will include the transaction count and amount for each type of transaction. To disable automatic close please access the Merchant Resource Centre (https://esplus.moneris.com), go to the Admin menu item and then choose Store Settings; the Batch Close options are located on this page.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
ecr_number = "64000001"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set batchClose = server.CreateObject("Moneris.USBatchClose")
out.setRequest batchClose.formatRequest(ecr_number)
out.sendRequest
%>

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

<%
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Open Totals

Open Totals allows the merchant to retrieve details about all Credit Card transactions within the currently open Batch. The response will include the transaction count and amount for each type of transaction. Open Totals returns a similar response to the Batch Close without closing the current Batch.

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
ecr_number = "64000001"

Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"

Set openTotals = server.CreateObject("Moneris.USOpenTotals")
out.setRequest openTotals.formatRequest(ecr_number)
out.sendRequest
%>

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

<%
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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
%>

</BODY>
</HTML>
```

Recur Update

Recur Update allows a user to alter characteristics of a previously registered Recurring Billing transaction. This feature is commonly used to update a customer's credit card information and the number of times it is to be billed (recur). Only fields sent to the gateway will be updated. Please see Appendix A. Definition of Request Fields and Appendix D. Recur and Recur Update Fields for description of each of the fields.

Recur Update – Credit Card example

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USPurchase")

'======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'======================================
'Set optional Level 2 details
commcard_invoice = "Invoice 56789"
commcard_tax_amount = "0.08"
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount)

'======================================
'Recurring setup.
recur_type = "day"
start_now = "true"
start_date = "2006/12/14"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
purchase.setRecur recur_type, start_now, start_date, num_recurs, period, recur_amount


order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "RecurSuccess:  " & out.getRecurSuccess & "<br>"
%>

</BODY>
</HTML>
```

Recur Update – Pinless Debit example

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USPurchase")

'=======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'=======================================
'Set optional Level 2 details
commcard_invoice = "Invoice 56789"
commcard_tax_amount = "0.08"
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount)

'=======================================
'Recurring setup.
recur_type = "day"
start_now = "true"
start_date = "2006/12/14"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
purchase.setRecur recur_type, start_now, start_date, num_recurs, period, recur_amount

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
pan = "4242424242424242"
exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "RecurSuccess:  " & out.getRecurSuccess & "<br>"
%>

</BODY>
</HTML>
```

Recur Update – ACH example

```
<% @Language = "VBScript" %>
<% Response.buffer = true %>
<%
store_id = "monusqa002"
api_token = "qatoken"
host = "https://esplusqa.moneris.com/gateway_us/servlet/MpgRequest"
Set out = server.CreateObject("Moneris.USRequest")
out.initRequest store_id, api_token, host

Set purchase = server.CreateObject("Moneris.USPurchase")

'======================================
'Set optional Customer ID
cust_id = "Customer_identifier"
purchase.setCustId( cust_id )

'======================================
'Set optional Level 2 details
commcard_invoice = "Invoice 56789"
commcard_tax_amount = "0.08"
purchase.setCommcardInvoice( commcard_invoice )
purchase.setCommcardTaxAmount( commcard_tax_amount)

'======================================
'Recurring setup.
recur_type = "day"
start_now = "true"
start_date = "2006/12/14"
num_recurs = "4"
period = "1"
recur_amount = "14.00"
purchase.setRecur recur_type, start_now, start_date, num_recurs, period, recur_amount

order_id = "mjr-"&Day(Date)&Month(Date)&Year(Date)&"-"&Hour(Now)&Minute(Now)&Second(Now)
amount = "1.00"
'pan = "4242424242424242"
'exp_date = "0812"
crypt_type = "7"
out.setRequest purchase.formatRequest( order_id, amount, pan, exp_date, 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 "Bank Totals:  " & out.getBankTotals & "<br>"
Response.Write "Card Type:  " & out.getCardType & "<br>"
Response.Write "Reference Number:  " & out.getReferenceNum & "<br>"
Response.Write "Transaction ID:  " & out.getTransID & "<br>"
Response.Write "Auth Code:  " & out.getAuthCode & "<br>"
Response.Write "Transaction Time:  " & out.getTransTime & "<br>"
Response.Write "Transaction Date:  " & out.getTransDate & "<br>"
Response.Write "Complete:  " & out.getCompleteStatus & "<br>"
Response.Write "Timeout:  " & out.getTimedoutStatus & "<br>"
Response.Write "Ticket:  " & out.getTicket & "<br>"
Response.Write "RecurSuccess:  " & out.getRecurSuccess & "<br>"
%>

</BODY>
</HTML>
```