



Merchant Integration Guide

XML File Upload - v. 1.1.2

Table of Contents

| | | |
|------------|--|----|
| 1. | Introduction | 4 |
| 2. | System and Skill Requirements | 4 |
| 3. | What is the Process I will need to follow? | 4 |
| 4. | Transaction Types and Transaction Flow..... | 5 |
| 5. | What Information Do I Need to Include in a Transaction Request..... | 8 |
| 6. | Example Transaction..... | 9 |
| 7. | Vault Admin Transactions | 11 |
| 8. | Transaction with Customer and Order Details..... | 15 |
| 9. | Recurring Transaction | 20 |
| 10. | How do I send a Transaction?..... | 24 |
| 11. | What Information will I get as a Response to My Transaction Request? | 24 |
| 12. | Filename Conventions..... | 25 |
| 13. | How Do I Test My Solution? | 25 |
| 14. | How Do I Activate My Store? | 26 |
| 15. | How Do I Get Help? | 26 |
| 16. | Appendix A. Definitions of Required Fields | 27 |
| 17. | Appendix B. Definitions of Response Fields..... | 29 |
| 18. | Appendix C. Definitions of Response Fields for Vault Transactions..... | 30 |
| 19. | Appendix D. Error Messages | 31 |
| 20. | Appendix E. Example | 32 |

****** 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. Introduction

By definition, a batch file may be used to forward large groups of transactions for processing. Merchants may forward multiple transaction types for multiple card plans in a single batch for processing, provided that the merchant is registered for the card plans included in the file.

The XML Batch File option refers to the ability to upload batch files where the fields are presented in an XML format.

There are a number of pieces of information that must be included in the batch file in a specific order. The following outlines the fields and the order in which they are expected for each transaction type. The last section of this document describes the information that will be provided back to you in the response file of processed batch files.

2. System and Skill Requirements

Before you start you will need to:

1. Have an SFTP client software which must use SSH2
2. Create the Batch File, must have the ".in" or ".xml" extension

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.

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

You will need to follow these steps.

1. Do the required development as outlined in this document
2. Test your solution in the test environment

Sending your batch file:

1. Establish an SFTP session with Moneris' SFTP server host using your SFTP client software
2. Log in using your Batch account username and password (please refer to NOTE)
3. Upload the Batch File
4. Wait approximately 2 seconds per transaction + 30 minutes or longer for processing
5. Retrieve your response file from the /out directory

**NOTE**

To receive your Production SFTP Username and Password, please have your main contact person e-mail the eSELECTplus Integration Support department.

E-mail: eselectplus@moneris.com

When e-mailing, please provide your merchant name, store ID, as well as your business address, phone and fax number. Your SFTP account information will be sent via fax to the number provided.

4. Transaction Types and Transaction Flow

eSELECTplus supports a wide variety of transactions through Batch File Upload. Below is a list of transactions supported, 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 – (authorization / preauthorization) 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.

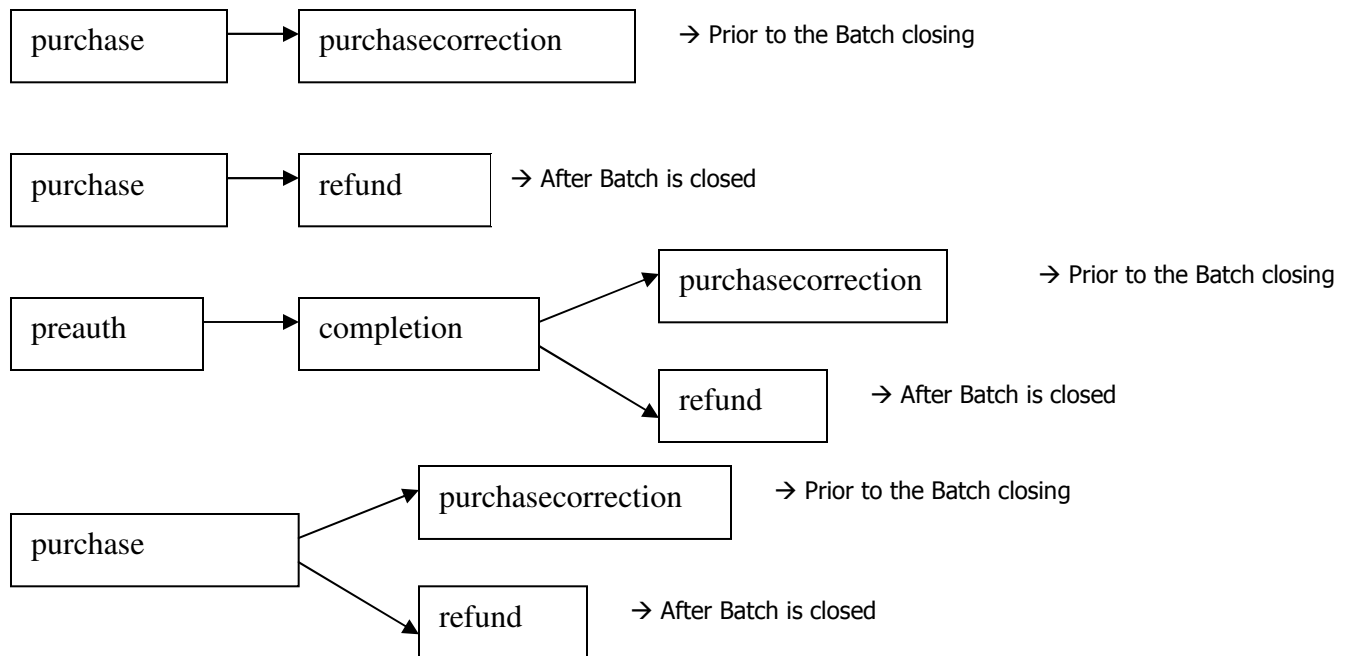
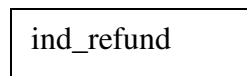
Capture – (Completion / PreAuth Completion) Once a PreAuth is obtained the funds that are locked need to be retrieved from the customer's credit card. The Capture retrieves the locked funds and readies them for settlement into the merchant's account.

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.

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

Process Flow for Basic Transactions**Transactions with no Follow-on Required**

Vault Admin Transactions

Add CC Vault Profile (res_add_cc) – Create a new credit card profile. The fields which may be sent in are outlined in the transaction examples which can be found in section 7 of this documentation.

Add Update Vault Profile (res_update_cc) – Update an existing credit card profile. The fields which may be updated are outlined in the transaction examples which can be found in section 7 of this documentation.

Delete Vault Profile (res_delete) – Delete an existing profile of any payment type using the unique data_key which was assigned when the profile was first added. The fields which may be sent in are outlined in the transaction examples which can be found in section 7 of this documentation.

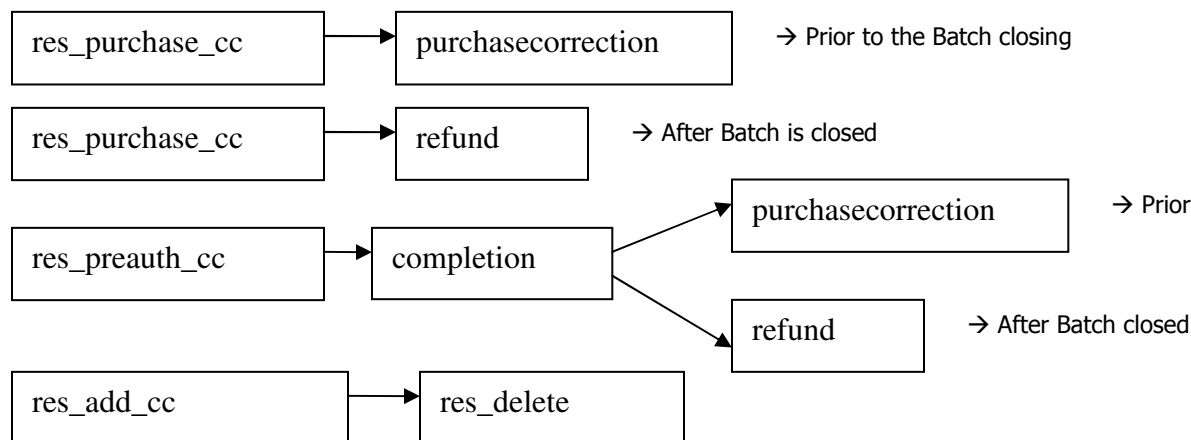
Vault Financial Transactions

Vault CC PreAuth (res_preauth_cc) – This is a Preauthorization transaction for Credit Card profiles only. This transaction will use a unique data_key which will identify a previously registered Credit Card profile. The details within the profile will be submitted to perform the Preauthorization transaction.

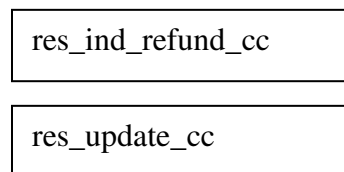
Vault CC Purchase (res_purchase_cc) - This is a Purchase transaction for Credit Card profiles only. This transaction will use a unique data_key which will identify a previously registered Credit Card profile. The details within the profile will be submitted to perform the Purchase transaction.

Vault CC Independent Refund (res_ind_refund_cc) - This is an independent refund transaction for Credit Card profiles only. This transaction will use a unique data_key which will identify a previously registered Credit Card profile. The details within the profile will be submitted to perform the refund transaction.

Process Flow for Vault Transactions



Transactions with no Follow-on Required



5. What Information Do I Need to Include in a Transaction Request

A transaction request contains various fields depending on the type of transaction that is being performed. The following table illustrates the fields that you can send for each transaction type. For a full description of each field please refer to Appendix A. Definitions of Required Fields. Please note that the fields must be in the correct order and the field names must be exactly the same. For an example of all of the transaction types and their respective structures, please refer to Appendix E. Example.

| Required Fields | |
|------------------|--------|
| Transaction Type | Fields |

Basic Transaction Types

| | |
|--------------------|---|
| Purchase | store_id, api_token, order_id, cust_id(optional), amount, pan, expdate, crypt_type |
| Pre-authorization | store_id, api_token, order_id, cust_id(optional), amount, pan, expdate, crypt_type |
| Capture | store_id, api_token, order_id (from preauth), comp_amount, txn_number(from preauth), crypt_type |
| Void | store_id, api_token, order_id(from orig txn), txn_number(from completion or purchase), crypt_type |
| Refund | store_id, api_token, order_id(from orig txn), amount, txn_number(from completion or purchase), crypt_type |
| Independent Refund | store_id, api_token, order_id, cust_id(optional), amount, pan, expdate, crypt_type |

Vault Initiated Financial Transactions (Basic)*

| | |
|---------------------|--|
| Vault CC Purchase | store_id, api_token, order_id, cust_id, amount, data_key, crypt_type |
| Vault CC PreAuth | store_id, api_token, order_id, cust_id, amount, data_key, crypt_type |
| Vault CC Ind Refund | store_id, api_token, order_id, cust_id, amount, data_key, crypt_type |

* Check Appendix A. Definitions of Required Fields for field definitions. Please refer to section 7 for the Vault Admin Transactions.



NOTE

The amount must have two decimal places and please do not include the \$ sign. Also, if it is less than 1.00 it must contain a leading 0 (ex. 0.10). Minimum amount is 0.01 and maximum is 9999999.99.

The format for expdate is **YYMM**. Please note that this is different from what appears on the credit card.

In addition to the required information you may send customer and invoice type information for future reference. The following table outlines the optional information you can send. For a list of all fields included within cust_info, please refer to section 8.

| Optional Fields | |
|---------------------------------------|--|
| Transaction Type | Fields |
| Purchase or Vault CC Purchase | cust_info – all fields contained within cust_info are required recur – all fields contained within recur are required |
| Pre-authorization or Vault CC PreAuth | cust_info – all fields contained within cust_info are required |
| Capture | n/a |

Void n/a
 Refund n/a
 Independent Refund or n/a
 Vault CC Ind Refund

Please note that the cust_info fields are not used for any type of address verification or fraud check.

6. Example Transaction

```
<?xml version="1.0"?>
<request>
  <store_id>store1</store_id>
  <api_token>yesguy</api_token>
  <purchase>
    <order_id>test_xml_00001</order_id>
    <cust_id>Customer Name</cust_id>
    <amount>10.00</amount>
    <pan>5454545454545454</pan>
    <expdate>0605</expdate>
    <crypt_type>1</crypt_type>
  </purchase>
  <purchase>
    <order_id>test_xml_00002</order_id>
    <amount>1.04</amount>
    <pan>4242424242424242</pan>
    <expdate>0806</expdate>
    <crypt_type>1</crypt_type>
  </purchase>
  <res_purchase_cc>
    <data_key>RbFbmySFugYNRSmTan6Gkm08r</data_key>
    <order_id>Moneris_test_Dec20_6</order_id>
    <cust_id>My_Customer_Name</cust_id>
    <amount>55.00</amount>
    <crypt_type>7</crypt_type>
  </res_purchase_cc>
</request>
```

Corresponding Example Response

```
<?xml version="1.0"?>
<response>
  <receipt>
    <ReceiptId>test_xml_00001</ReceiptId>
    <ReferenceNum>660021730013780180</ReferenceNum>
    <ResponseCode>027</ResponseCode>
    <ISO>01</ISO>
    <AuthCode>009213</AuthCode>
    <TransTime>16:04:33</TransTime>
    <TransDate>2006-06-20</TransDate>
    <TransType>00</TransType>
    <Complete>true</Complete>
    <Message>APPROVED *                               =</Message>
    <TransAmount>10.00</TransAmount>
    <CardType>M</CardType>
    <TransID>84112-18-0</TransID>
    <TimedOut>false</TimedOut>
    <BankTotals>null</BankTotals>
    <Ticket>null</Ticket>
  </receipt>
  <receipt>
    <ReceiptId>test_xml_00002</ReceiptId>
    <ReferenceNum>660021630014070190</ReferenceNum>
    <ResponseCode>075</ResponseCode>
    <ISO>14</ISO>
    <AuthCode>000000</AuthCode>
    <TransTime>16:04:35</TransTime>
    <TransDate>2006-06-20</TransDate>
```

```
<TransType>00</TransType>
<Complete>true</Complete>
<Message>DECLINED * CARD PROBLEM =CARD PROBLEM</Message>
<TransAmount>1.04</TransAmount>
<CardType>00</CardType>
<TransID>156091-19-0</TransID>
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
</receipt>
<receipt>
  <DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
  <ReceiptId>Moneris_test_Dec20_6</ReceiptId>
  <ReferenceNum>660021820010918800</ReferenceNum>
  <ResponseCode>027</ResponseCode>
  <ISO>01</ISO>
  <AuthCode>001945</AuthCode>
  <Message>APPROVED * =</Message>
  <TransTime>14:03:53</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>00</TransType>
  <Complete>true</Complete>
  <TransAmount>55.00</TransAmount>
  <CardType>M</CardType>
  <TransID>966830-0_7</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>false</CorporateCard>
  <RecurSuccess>false</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
  <ResolveData>
    <cust_id>My_Customer_Name</cust_id>
    <phone>555-800-1122</phone>
    <email>my_email@mail.com</email>
    <note>I have no note at this time</note>
    <expdate>1201</expdate>
    <avs_street_number>1</avs_street_number>
    <avs_street_name>main st</avs_street_name>
    <avs_zipcode>123456</avs_zipcode>
    <masked_pan>5454***5454</masked_pan>
    <crypt_type>7</crypt_type>
  </ResolveData>
</receipt>
</response>
```

7. Vault Admin Transactions

Included below is the sample code for the Administrative transactions. These Administrative transactions allow the user to perform such tasks as creating new Vault profiles, updating existing profiles and deleting existing profiles.

res_add_cc – creates a new credit card profile.

res_update_cc – Update an existing profile using the data_key which was assigned when the profile was first added. This transaction allows portions of the profile to be updated.

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

| Transaction Type | Fields |
|------------------|---|
| res_add_cc | store_id, api_token, order_id, cust_id(optional), phone(optional), email(optional), note(optional), pan, exp_date, crypt_type |
| res_update_cc | store_id, api_token, order_id, data_key, cust_id(optional), phone(optional), email(optional), note(optional), pan(optional), exp_date(optional), crypt_type(optional) |
| res_delete | store_id, api_token, order_id, data_key |

Note: Please refer to Appendix A. Definitions of Required Fields for field definitions.

Examples

```
<?xml version="1.0"?>
<request>
<store_id>store3</store_id>
<api_token>yesguy</api_token>
<res_add_cc>
  <order_id>Moneris_test_Dec20_1</order_id>
  <pan>4242424242424242</pan>
  <expdate>1212</expdate>
  <crypt_type>7</crypt_type>
</res_add_cc>
<res_add_cc>
  <order_id>Moneris_test_Dec20_2</order_id>
  <pan>5454545454545454</pan>
  <expdate>1201</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
</res_add_cc>
<res_update_cc>
  <order_id>Moneris_test_Dec20_3</order_id>
  <data_key>XKwBrc6n8sHIJftKlmNf0TFb3</data_key>
  <expdate>1212</expdate>
</res_update_cc>
<res_update_cc>
  <order_id>Moneris_test_Dec20_4</order_id>
  <data_key>RbFbmySFugYNRSmTan6Gkm08r</data_key>
  <pan>5454545454545454</pan>
  <expdate>1201</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
</res_update_cc>
<res_delete>
  <order_id>Moneris_test_Dec20_5</order_id>
  <data_key>RbFbmySFugYNRSmTan6Gkm08r</data_key>
</res_delete>
</request>
```

Example 1 processes an Add CC Vault Profile transaction which registers the new credit card profile. In this example only the mandatory fields have been provided.

Example 2 processes an Add CC Vault Profile transaction which registers the new credit card profile. In this example all of the optional profile details have been provided: cust_id, phone, email, note

Example 3 processes an Update CC Vault Profile transaction which updates an existing profile. In this example, only one field will be updated: expdate

Example 4 processes an Update CC Vault Profile transaction which updates an existing profile. In this example, all of the fields that have been submitted will be updated. In the response, please note the AVS details that have also been included. The ResolveData portion of the XML will return what is currently stored within the profile. In this particular example, the profile already contains the three AVS fields.

Example 5 processes a Delete Vault Profile transaction which will completely remove the profile and its details.

Corresponding Example Response

```
<?xml version="1.0"?>
<response>
<receipt>
  <DataKey>wjxVIIIFcJAqKhBZ3URAST08MO</DataKey>
  <ReceiptId>Moneris_test_Dec20_1</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>001</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <Message>Successfully registered CC details.</Message>
  <TransTime>14:02:19</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <TransAmount>null</TransAmount>
  <CardType>null</CardType>
  <TransID>null</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>null</CorporateCard>
  <RecurSuccess>null</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
  <ResolveData>
    <cust_id></cust_id>
    <phone></phone>
    <email></email>
    <note></note>
    <crypt_type>7</crypt_type>
    <masked_pan>4242***4242</masked_pan>
    <expdate>1212</expdate>
  </ResolveData>
</receipt>
<receipt>
  <DataKey>NrFoDGGF6QqtDGVKiVm6etwp3</DataKey>
  <ReceiptId>Moneris_test_Dec20_2</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>001</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <Message>Successfully registered CC details.</Message>
  <TransTime>14:02:29</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <TransAmount>null</TransAmount>
  <CardType>null</CardType>
```

```
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
  <crypt_type>7</crypt_type>
  <masked_pan>5454***5454</masked_pan>
  <expdate>1201</expdate>
</ResolveData>
</receipt>
<receipt>
  <DataKey>XKwBrc6n8sHIJftKlmNf0TFb3</DataKey>
  <ReceiptId>Moneris_test_Dec20_3</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>001</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <Message>Successfully updated CC details.</Message>
  <TransTime>14:02:39</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <TransAmount>null</TransAmount>
  <CardType>null</CardType>
  <TransID>null</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>null</CorporateCard>
  <RecurSuccess>null</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
  <ResolveData>
    <cust_id></cust_id>
    <phone></phone>
    <email></email>
    <note></note>
    <crypt_type>7</crypt_type>
    <expdate>1212</expdate>
    <masked_pan>4242***4242</masked_pan>
  </ResolveData>
</receipt>
<receipt>
  <DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
  <ReceiptId>Moneris_test_Dec20_4</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>001</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <Message>Successfully updated CC details.</Message>
  <TransTime>14:02:49</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <TransAmount>null</TransAmount>
  <CardType>null</CardType>
  <TransID>null</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>null</CorporateCard>
  <RecurSuccess>null</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
```

```
<ResolveData>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
  <avs_street_name>main st</avs_street_name>
  <avs_street_number>1</avs_street_number>
  <avs_zipcode>123456</avs_zipcode>
  <crypt_type>7</crypt_type>
  <expdate>1201</expdate>
  <masked_pan>5454***5454</masked_pan>
</ResolveData>
</receipt>
<receipt>
  <DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
  <ReceiptId>Moneris_test_Dec20_5</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>001</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <Message>Successfully deleted CC details.</Message>
  <TransTime>16:44:20</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <TransAmount>null</TransAmount>
  <CardType>null</CardType>
  <TransID>null</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>null</CorporateCard>
  <RecurSuccess>null</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
  <ResolveData>
    <cust_id>My_Customer_Name</cust_id>
    <phone>555-800-1122</phone>
    <email>my_email@mail.com</email>
    <note>I have no note at this time</note>
    <expdate>1201</expdate>
    <avs_street_number>1</avs_street_number>
    <avs_street_name>main st</avs_street_name>
    <avs_zipcode>123456</avs_zipcode>
    <masked_pan>5454***5454</masked_pan>
    <crypt_type>7</crypt_type>
  </ResolveData>
</receipt>
</response>
```

8. Transaction with Customer and Order Details

You can also upload extra customer and order details with a basic Purchase or Pre-Authorization transaction via the XML file upload. When uploading extra details all of the fields within cust_info must be included.

The format of the request contains the following fields (examples follow):

| Field Definitions | | |
|--|-------------|---|
| Field Name | Size/Type | Description |
| purchase preauth res_purchase_cc res_preauth_cc (trans_type) | | All transactions with extra customer and order details must be sent with trans_type as 'purchase', 'preauth', 'res_purchase_cc', or 'res_preauth_cc'. The trans_type is case sensitive and must be all in lower case. |
| order_id | 50 / alpha | Alphanumeric – must be unique for every order |
| cust_id | 50 / an | This is an optional field that can be sent as part of a Purchase or PreAuth request. It is searchable from the Moneris Merchant Resource Centre. It is commonly used for policy number, membership number, student ID or invoice number. Please note, for Vault transactions if you omit the customer ID, then the cust_id stored within the Vault profile will be submitted. Otherwise, if you pass the cust_id field in the transaction, then it will be submitted for this transaction only. |
| amount | 9 / decimal | Amount of the transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |
| data_key | 50 / an | An alphanumeric identifier used in Vault transactions to uniquely identify a Vault profile. The data_key is generated by Moneris Solutions and returned to the merchant when the profile is first registered using the res_add_cc transaction. This field is mandatory for Vault transactions. |
| pan | 20 / num | Credit Card number – no spaces, no dashes (numeric only). This field is mandatory for non-vault transactions. |
| expdate | 4 / num | Format – YYMM. Please note, this is opposite from what appears on the card. This field is mandatory for non-vault transactions. |
| crypt_type | 1 / num | E-Commerce Indicator: 1 - Mail Order/Telephone Order – Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment Please refer to Appendix A. Definitions of Required Fields for a list of all Crypt Types. |

Billing and Shipping Information

NOTE: The fields for billing and shipping information are identical and must all be sent within the <billing> and <shipping> XML tags.

| | |
|--------------|---------|
| first_name | 30 / an |
| last_name | 30 / an |
| company_name | 30 / an |
| address | 30 / an |
| city | 30 / an |

| | |
|---------------|---------|
| province | 30 / an |
| postal_code | 30 / an |
| country | 30 / an |
| phone_number | 30 / an |
| fax | 30 / an |
| tax1 | 30 / an |
| tax2 | 30 / an |
| tax3 | 30 / an |
| shipping_cost | 30 / an |

Item Information

NOTE: The fields for item information must all be sent within the <item> XML tag. You may send multiple items - please refer to example below.

| | | |
|-----------------|-------------|---|
| name | 15 / an | |
| quantity | 4 / num | You must send a quantity > 0 or the item will not be added to the item list (i.e. minimum 1, maximum 9999) |
| product_code | 10 / an | |
| extended_amount | 9 / decimal | This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |

Extra Details

| | |
|--------------|---------|
| email | 50 / an |
| instructions | 50 / an |



NOTE

The data sent in Billing and Shipping Address fields will not be used for any address verification.

Example

```
<?xml version="1.0"?>
<request>
<store_id>store1</store_id>
<api_token>yesguy</api_token>
<purchase>
  <order_id>Moneris_test_Nov9_custinfo</order_id>
  <amount>10.00</amount>
  <pan>4242424242424242</pan>
  <expdate>0901</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <cust_info>
    <billing>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
    </billing>
  </cust_info>
</purchase>
</request>
```



```
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</billing>
<shipping>
  <first_name>Bob</first_name>
  <last_name>Smith</last_name>
  <company_name>None</company_name>
  <address>101 Main St</address>
  <city>Springfield</city>
  <province>YU</province>
  <postal_code>Z1Z1Z1</postal_code>
  <country>CAD</country>
  <phone_number>919-555-5555</phone_number>
  <fax>919-555-5550</fax>
  <tax1>0.10</tax1>
  <tax2>0.20</tax2>
  <tax3>0.00</tax3>
  <shipping_cost>0.00</shipping_cost>
</shipping>
<email>bob@smith.com</email>
<instructions>Please deliver to back door</instructions>
<item>
  <name>Red Shoes</name>
  <quantity>1</quantity>
  <product_code>shoes_101</product_code>
  <extended_amount>1.00</extended_amount>
</item>
<item>
  <name>Blue Suede Shoes</name>
  <quantity>10</quantity>
  <product_code>shoes_102</product_code>
  <extended_amount>3.00</extended_amount>
</item>
</cust_info>
</purchase>
<res_purchase_cc>
  <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
  <order_id>Moneris_test_Dec20_6</order_id>
  <cust_id>My_Customer_Name</cust_id>
  <amount>55.00</amount>
  <crypt_type>7</crypt_type>
  <cust_info>
    <billing>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </billing>
    <shipping>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
```

```

        <tax3>0.00</tax3>
        <shipping_cost>0.00</shipping_cost>
    </shipping>
    <email>bob@smith.com</email>
    <instructions>Please deliver to back door</instructions>
    <item>
        <name>Red Shoes</name>
        <quantity>1</quantity>
        <product_code>shoes_101</product_code>
        <extended_amount>1.00</extended_amount>
    </item>
    <item>
        <name>Blue Suede Shoes</name>
        <quantity>10</quantity>
        <product_code>shoes_102</product_code>
        <extended_amount>3.00</extended_amount>
    </item>
    </cust_info>
</res_purchase_cc>
</request>

```

The above example submits a Purchase transaction with all optional fields: cust_id as well as cust_info. All fields within cust_info have been populated and 2 items have been included.

Corresponding Example Response

```

<?xml version="1.0"?>
<response>
    <receipt>
        <ReceiptId>Moneris_test_Nov9_custinfo</ReceiptId>
        <ReferenceNum>640000030010120330</ReferenceNum>
        <ResponseCode>001</ResponseCode>
        <ISO></ISO>
        <AuthCode>756311</AuthCode>
        <TransTime>18:55:17</TransTime>
        <TransDate>2006-11-09</TransDate>
        <TransType>00</TransType>
        <Complete>true</Complete>
        <Message> APPROVED *                               =</Message>
        <TransAmount>10.42</TransAmount>
        <CardType>V</CardType>
        <TransID>1297-0-1</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
    </receipt>
    <receipt>
        <DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
        <ReceiptId>Moneris_test_Dec20_6</ReceiptId>
        <ReferenceNum>660021820010918800</ReferenceNum>
        <ResponseCode>027</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>001945</AuthCode>
        <Message>APPROVED *                               =</Message>
        <TransTime>14:03:53</TransTime>
        <TransDate>2010-12-21</TransDate>
        <TransType>00</TransType>
        <Complete>true</Complete>
        <TransAmount>55.00</TransAmount>
        <CardType>M</CardType>
        <TransID>966830-0_7</TransID>
        <TimedOut>false</TimedOut>
        <CorporateCard>false</CorporateCard>
        <RecurSuccess>false</RecurSuccess>
        <AvsResultCode>null</AvsResultCode>
        <CvdResultCode>null</CvdResultCode>
        <ResSuccess>true</ResSuccess>
        <PaymentType>cc</PaymentType>
        <ResolveData>

```

```
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<expdate>1201</expdate>
<avs_street_number>1</avs_street_number>
<avs_street_name>main st</avs_street_name>
<avs_zipcode>123456</avs_zipcode>
<masked_pan>5454***5454</masked_pan>
<crypt_type>7</crypt_type>
  </ResolveData>
</receipt>
</response>
```

9. Recurring Transaction

You can also upload recurring transactions via the XML file upload. When uploading a recurring transaction you will need to indicate the recurring amount, interval and start date and the number of time it is to recur. You can also optionally bill a different amount immediately.

The recurring request can only be sent as a purchase. The format of the request contains the following fields (examples follow):

| Field Definitions | | |
|---|---------------------|---|
| Field Name | Size/Type | Description |
| purchase res_purchase_cc (trans_type) | | All recurs must be sent with trans_type as 'purchase' or 'res_purchase_cc'. The trans_type is case sensitive and must be all in lower case. |
| order_id | 50 / alpha | Alphanumeric – must be unique for every order |
| amount (charge now) | 9 / decimal | When start_now is set to 'true' the amount field becomes the amount to be billed immediately. When start_now is set to 'false' the amount field should be the same as the recur_amount field. The amount can be a minimum of 0.01 up to a maximum of 9999999.99. It must contain 2 decimal places. |
| data_key | 50 / an | An alphanumeric identifier used in Vault transactions to uniquely identify a Vault profile. The data_key is generated by Moneris Solutions and returned to the merchant when the profile is first registered using the res_add_cc transaction. This field is mandatory for Vault transactions. |
| pan | 20 / num | Credit Card number – no spaces, no dashes (numeric only). This field is mandatory for non-vault transactions. |
| expdate | 4 / num | Format – YYMM Please note, this is opposite from what appears on the card. This field is mandatory for non-vault transactions. |
| crypt_type | 1 / num | E-Commerce Indicator: 2 - Mail Order/Telephone Order - Recurring Please refer to Appendix A. Definitions of Required Fields for a list of all Crypt Types. |
| recur_unit | day, week, month | The unit that you wish to use as a basis for the Interval. This can be set as day, week or month. Then using the "period" field you can configure how many days, weeks, months between billing cycles. |
| period | 0 – 999/num | This is the number of recur_units you wish to pass between billing cycles. Example : period = 3, recur_unit=month -> Card will be billed every 3 months. period = 4, recur_unit=weeks -> Card will be billed every 4 weeks. period = 45, recur_unit=day -> Card will be billed every 45 days. Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future. |
| start_date | YYYY/MM/DD | This is the date on which the first charge will be billed. The value must be in the future. It cannot be the day on which the transaction is being sent. If the transaction is to be billed immediately the start_now feature must be set to true and the start_date should be set at the desired interval after today. |
| start_now | true / false | When a charge is to be made against the card immediately start_now should be set to 'true'. If the billing is to start in the future then this value is to be set to 'false'. When start_now is set to 'true' the amount to be billed immediately may differ from the recur amount billed on a regular |

| | | |
|--------------|--------------|---|
| | | basis thereafter. |
| recur_amount | 9 / decimal | Amount of the recurring transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99. This is the amount that will be billed on the start_date and every interval thereafter. |
| num_recurs | 1 – 99 / num | The number of times to recur the transaction. |

**NOTE**

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

Example

```
<?xml version="1.0"?>
<request>
  <store_id>store1</store_id>
  <api_token>yesguy</api_token>
  <purchase>
    <order_id>464571415531_1</order_id>
    <amount>3.00</amount>
    <pan>4242424242424242</pan>
    <expdate>0803</expdate>
    <crypt_type>7</crypt_type>
    <recur>
      <recur_unit>week</recur_unit>
      <start_now>true</start_now>
      <start_date>2011/10/30</start_date>
      <num_recurs>4</num_recurs>
      <period>2</period>
      <recur_amount>1.00</recur_amount>
    </recur>
  </purchase>
  <purchase>
    <order_id>464571415532_2</order_id>
    <amount>1.00</amount>
    <pan>4242424242424242</pan>
    <expdate>0903</expdate>
    <crypt_type>7</crypt_type>
    <recur>
      <recur_unit>day</recur_unit>
      <start_now>false</start_now>
      <start_date>2011/09/15</start_date>
      <num_recurs>10</num_recurs>
      <period>30</period>
      <recur_amount>1.00</recur_amount>
    </recur>
  </purchase>
  <purchase>
    <order_id>464571415533_3</order_id>
    <amount>5.00</amount>
    <pan>4242424242424242</pan>
    <expdate>0703</expdate>
    <crypt_type>7</crypt_type>
    <recur>
      <recur_unit>month</recur_unit>
      <start_now>true</start_now>
      <start_date>2011/11/11</start_date>
      <num_recurs>12</num_recurs>
      <period>1</period>
      <recur_amount>1.00</recur_amount>
    </recur>
  </purchase>
</request>
```

```

    </recur>
  </purchase>
  <res_purchase_cc>
    <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
    <order_id>Moneris_test_Dec20_4</order_id>
    <cust_id>My_Customer_Name</cust_id>
    <amount>55.00</amount>
    <crypt_type>7</crypt_type>
    <recur>
      <recur_unit>month</recur_unit>
      <start_now>true</start_now>
      <start_date>2011/12/01</start_date>
      <num_rekurs>12</num_rekurs>
      <period>1</period>
      <recur_amount>1.00</recur_amount>
    </recur>
  </res_purchase_cc>
</request>

```

Example 1 will result in the card being billed immediately for \$3.00 and then starting on the 30th of Oct it will be billed \$1.00 every 2 weeks, 4 times. Final result will be that the card is charged 5 times – once for 3.00, 4 for 1.00.

Example 2 will result in the card being billed starting the 15th and every 30 days thereafter, 10 times

Example 3 will result in the card being billed immediately for \$5.00 and then starting on the 11th of Nov it will be billed 1.00 every month for 12 months. The card will be billed a total of 13 times.

Example 4 will result in the credit card associated with that Vault profile (data_key) being billed immediately for \$55.00 and then starting on the 1st of Dec it will be billed 1.00 every month for 12 months. The card will be billed a total of 13 times.

Corresponding Example Response

```

<?xml version="1.0"?>
<response>
<receipt>
  <ReceiptId>464571415531_1</ReceiptId>
  <ReferenceNum>660021630014070250</ReferenceNum>
  <ResponseCode>027</ResponseCode>
  <ISO>01</ISO>
  <AuthCode>009250</AuthCode>
  <TransTime>16:24:34</TransTime>
  <TransDate>2006-06-20</TransDate>
  <TransType>00</TransType>
  <Complete>true</Complete>
  <Message>APPROVED *                               =: Recurring transaction successfully
  registered.</Message>
  <TransAmount>3.00</TransAmount>
  <CardType>V</CardType>
  <TransID>156097-25-0</TransID>
  <TimedOut>false</TimedOut>
  <BankTotals>null</BankTotals>
  <Ticket>null</Ticket>
  <RecurSuccess>true</RecurSuccess>
</receipt>
<receipt>
  <ReceiptId>464571415532_2</ReceiptId>
  <ReferenceNum>null</ReferenceNum>
  <ResponseCode>null</ResponseCode>
  <ISO>null</ISO>
  <AuthCode>null</AuthCode>
  <TransTime>null</TransTime>
  <TransDate>null</TransDate>
  <TransType>null</TransType>
  <Complete>true</Complete>
  <Message>Recurring transaction successfully registered.</Message>
  <TransAmount>null</TransAmount>

```

```
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
<RecurSuccess>true</RecurSuccess>
</receipt>
<receipt>
  <ReceiptId>464571415533_3</ReceiptId>
  <ReferenceNum>660021730013780250</ReferenceNum>
  <ResponseCode>027</ResponseCode>
  <ISO>01</ISO>
  <AuthCode>010212</AuthCode>
  <TransTime>16:24:41</TransTime>
  <TransDate>2006-06-20</TransDate>
  <TransType>00</TransType>
  <Complete>true</Complete>
  <Message>APPROVED *           =: Recurring transaction successfully registered.</Message>
  <TransAmount>5.00</TransAmount>
  <CardType>V</CardType>
  <TransID>84119-25-0</TransID>
  <TimedOut>false</TimedOut>
  <BankTotals>null</BankTotals>
  <Ticket>null</Ticket>
  <RecurSuccess>true</RecurSuccess>
</receipt>
<receipt>
  <DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
  <ReceiptId>Moneris_test_Dec20_4</ReceiptId>
  <ReferenceNum>660021820010918800</ReferenceNum>
  <ResponseCode>027</ResponseCode>
  <ISO>01</ISO>
  <AuthCode>001945</AuthCode>
  <Message>APPROVED *           =: Recurring transaction successfully registered.</Message>
  <TransTime>14:03:53</TransTime>
  <TransDate>2010-12-21</TransDate>
  <TransType>00</TransType>
  <Complete>true</Complete>
  <TransAmount>55.00</TransAmount>
  <CardType>M</CardType>
  <TransID>966830-0_7</TransID>
  <TimedOut>false</TimedOut>
  <CorporateCard>false</CorporateCard>
  <RecurSuccess>true</RecurSuccess>
  <AvsResultCode>null</AvsResultCode>
  <CvdResultCode>null</CvdResultCode>
  <ResSuccess>true</ResSuccess>
  <PaymentType>cc</PaymentType>
  <ResolveData>
    <cust_id>My_Customer_Name</cust_id>
    <phone>555-800-1122</phone>
    <email>my_email@mail.com</email>
    <note>I have no note at this time</note>
    <expdate>1201</expdate>
    <avs_street_number>1</avs_street_number>
    <avs_street_name>main st</avs_street_name>
    <avs_zipcode>123456</avs_zipcode>
    <masked_pan>5454**5454</masked_pan>
    <crypt_type>7</crypt_type>
  </ResolveData>
</receipt>
</response>
```

10. How do I send a Transaction?

We have an SFTP account setup in our test environment that you can use to upload test data for processing. To upload the file you will need to connect to:

| | Test Environment | Production |
|-------------|------------------|---|
| Host | esqa.moneris.com | www3.moneris.com |
| Port | 22 | 22 |
| Server Type | SFTP using SSH2 | SFTP using SSH2 |
| Username | store3 | Provided by Moneris (please refer to NOTE) |
| Password | store3 | Provided by Moneris (please refer to NOTE) |

To receive your Production SFTP Username and Password, please have your main contact person e-mail the eSELECTplus Integration Support department.

**NOTE**

E-mail: eselectplus@moneris.com

When e-mailing, please provide your merchant name, store ID, as well as your business address, phone and fax number. Your SFTP account information will be sent via fax to the number provided.

11. What Information will I get as a Response to My Transaction Request?

For each transaction you will receive a response message. For a full description of each field please refer to Appendix B. Definitions of Response Fields. After the file is successfully processed go to the /out directory of the SFTP account and download the .out file. This will contain all the transaction responses. Please review the response file for all of the transactions.

To determine whether a transaction is successful or not the field that must be checked is ResponseCode. See the table below to determine the transaction result.

| Response Code | Result |
|----------------------|------------|
| 0 – 49 (inclusive) | Approved |
| 50 – 999 (inclusive) | Declined |
| null | Incomplete |

For a full list of response codes and the associated message please refer to <https://www.eselectplus.ca/en/downloadable-content> and download the Response Code document.

12. Filename Conventions

When you are uploading a file the filename must conform to certain standards. The file name must be alphanumeric. It cannot contain any spaces. The extension must be ".xml" and it must be lowercase. Filenames that do not meet these requirements will not be processed.

When uploading a file it must be put in your default root directory. Do not put it in the processing directory – it will not get processed. When a file is complete the response file will be put in the /out folder. The filename will be identical to what was uploaded but will have an extension ".out"

| Request File | Response File |
|---------------------|-------------------------|
| testing_june_20.xml | testing_june_20.xml.out |

13. How Do I Test My Solution?

A testing environment is available for you to connect to while you are integrating your site to our payment gateway. The test environment is generally available 7x24, however since it is a test environment we cannot guarantee 100% availability. Also, please be aware that other merchants are using the test environment so you may see transactions and batch files that you did not create. As a courtesy to others that are testing we ask that when you are processing Refunds, downloading/deleting batch files and/or trying other functions, that you use only the transactions/files that you created.

When using the XML Batch file upload in the test environment you will need to use test store_id and api_token. These are different than your production IDs. The IDs that you can use in the test environment are in the table below. To upload .xml batch files to the test environment, please refer to section 10.

| Test IDs | | | |
|----------|-----------|----------|----------|
| store_id | api_token | Username | Password |
| store1 | yesguy | DemoUser | password |
| store2 | yesguy | DemoUser | password |
| store3 | yesguy | DemoUser | password |

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

| Test Card Numbers | |
|-------------------|------------------|
| Card Plan | Card Number |
| MasterCard | 5454545454545454 |
| Visa | 4242424242424242 |
| Amex | 373599005095005 |
| Diners | 36462462742008 |

To verify that these transactions have been properly received and processed, please refer to the .out file as well as logging into the Merchant Resource Centre to view reports of your transactions. To access the Merchant Resource Centre in the test environment go to <https://esqa.moneris.com/mpg> and use the logins provided in the table below. If you have uploaded the batch file to store3, please log into the Merchant Resource Centre into store3 to locate your transactions. These test store ID's, usernames and passwords are different than your production IDs.

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

The test environment will approve and decline transactions based on the penny value of the amount field.

For example, a transaction made for the amount of \$399.00 or \$1.00 will approve since the .00 penny value is set to approve in the test environment. Transactions in the test environment should not exceed \$1000.00. This limit does not exist in the production environment. For a list of all current test environment responses for various penny values, please see the Test Environment Penny Response table available at <https://www.eselectplus.ca/en/downloadable-content>.

**NOTE**

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

14. How Do I Activate My Store?

Once you have received your activation letter/fax go to <https://www3.moneris.com/connect/en/activate/index.php> as instructed in the letter/fax. You will need to input your store ID and merchant ID then click on 'Activate'. In this process you will need to create an administrator account that you will use to log into the Merchant Resource Centre to access and administer your eSELECTplus store. You will need to use this Store ID and API Token to send transactions via the SFTP Batch File Upload.

Once you have created your first Merchant Resource Centre user, please log on to the Interface by clicking the "eSELECTplus" button. Once you have logged in please proceed to ADMIN and then STORE SETTINGS. At the bottom please place a check beside Batch Upload. This will allow us to keep you up to date regarding any changes to the Batch Upload that may affect your store. Also, this is where you may locate your production API Token.

Next, please have your main contact person contact the eSELECTplus Integration Support department for your SFTP account username and password. The support team may be reached at 1-866-562-4354 or by e-mail at eselectplus@moneris.com. When e-mailing, please provide your merchant number, merchant name, store ID, as well as your business address, phone and fax number. Your SFTP account information will be sent via fax to the number provided.

15. How Do I Get Help?

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

Phone: 1-866-562-4354

Email: eselectplus@moneris.com

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

16. Appendix A. Definitions of Required Fields

| Required Fields | | |
|--|---------------|--|
| Variable Name | Size/Type | Description |
| Required Fields (for Basic Credit Card Transactions) | | |
| trans_type | an | The transaction type is a case sensitive field that must be all in lower case. The trans_type field must match one of these available transaction types: <ul style="list-style-type: none"> • purchase • preauth • completion • purchasecorrection • refund • ind_refund |
| order_id | 50 / an | Merchant defined unique transaction identifier - must be unique for every purchase, preauth and ind_refund attempt. For refund, completion and correction the order_id must reference the original transaction. |
| pan | 20 / variable | Credit Card Number - no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration for future expansion and/or potential support of private label card ranges. |
| expdate | 4 / num | Expiry Date - format YYMM no spaces or slashes. PLEASE NOTE THAT THIS IS REVERSED FROM THE DATE DISPLAYED ON THE PHYSICAL CARD WHICH IS MMY |
| amount | 9 / decimal | Amount of the transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |
| comp_amount | 9 / decimal | Amount of the follow-on Capture (Completion) transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |
| crypt_type | 1 / an | E-Commerce Indicator: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 - Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant 8 - Non Secure Transaction (Web or Email Based) 9 - SET non - Authenticated transaction |
| txn_number | 255 / varchar | Used when performing follow on transactions - this must be filled with the value that was returned as the Txn_number in the response of the original transaction. When performing a Capture this must reference the PreAuth. When performing a Refund or a Void this must reference the Capture or the Purchase. |
| cust_id | 50 / an | This is an optional field that can be sent as part of a Purchase or PreAuth request. It is searchable from the Moneris Merchant Resource Centre. It is commonly used for policy number, membership number, student ID or invoice number. |

Vault Fields

| | | |
|------------|---------|--|
| trans_type | an | <p>The transaction type is a case sensitive field that must be all in lower case. The trans_type field must match one of these available transaction types:</p> <ul style="list-style-type: none"> • res_add_cc • res_update_cc • res_delete • res_purchase_cc • res_preauth_cc • res_ind_refund_cc |
| data_key | 50 / an | An alphanumeric identifier used in Vault transactions to uniquely identify a Vault profile. The data_key is generated by Moneris Solutions and returned to the merchant when the profile is first registered using the res_add_cc transaction. |
| order_id | 50 / an | <p>Merchant defined unique transaction identifier. In a Vault admin transaction (res_add_cc, res_update_cc, res_delete) the Order ID is only used to match the request with the response. The Order ID will not be stored within the Moneris Merchant Resource Centre (MRC).</p> <p>In all Vault financial transactions (res_purchase_cc, res_preauth_cc, res_ind_refund_cc) the Order ID must be unique and will be used to reference the transaction within the MRC.</p> <p>Characters allowed for Order ID: a-z A-Z 0-9 _ - : . @ spaces</p> |
| cust_id | 50 / an | This is an optional field that can be either registered in a profile or sent as part of a res_purchase_cc, res_preauth_cc, or res_ind_refund_cc request. It is searchable from the Moneris Merchant Resource Centre. It is commonly used for policy number, membership number, student ID or invoice number. |
| phone | 30 / an | Phone number of the customer. This is an optional field which can be sent in when creating a Vault profile. |
| email | 30 / an | Email of the customer. This is an optional field which can be sent in when creating a Vault profile. |
| note | 30 / an | This field can be used for supplementary information which is to be sent in with the transaction. This is an optional field which can be sent in when creating a Vault profile. |

17. Appendix B. Definitions of Response Fields

| Response Fields | | |
|-----------------|------------|---|
| Variable Name | Size/Type | Description |
| ReceiptId | 50 / an | order_id specified in request |
| ReferenceNum | 18 / num | The reference number is an 18 character string that references the terminal used to process the transaction as well as the shift, batch and sequence number. This data is typically used to reference transactions on the host systems and must be displayed on any receipt presented to the customer. This information should be stored by the merchant. The following illustrates the breakdown of this field where "660123450010690030" is the reference number returned in the message, "66012345" is the terminal id, "001" is the shift number, "069" is the batch number and "003" is the transaction number within the batch. |
| ReponseCode | 3 / num | Moneris Host Transaction identifier. Transaction Response Code < 50: Transaction approved >= 50: Transaction declined NULL: Transaction was not sent for authorization * If you would like further details on the response codes that are returned please see the Response Codes document available at https://www.eselectplus.ca/en/downloadable-content |
| ISO | 2 / num | ISO response code |
| AuthCode | 8 / an | Authorization code returned from the issuing institution |
| TransTime | ##:##:## | Processing host time stamp |
| TransDate | yyyy-mm-dd | Processing host date stamp |
| TransType | an | Type of transaction that was performed 00 = Purchase, Vault CC Purchase 01 = Preauthorization, Vault CC Preauthorization 02 = Capture 04 = Refund, Independent Refund, Vault CC Ind Refund 11 = Void |
| Complete | true/false | Transaction was sent to authorization host and a response was received |
| Message | 100 / an | Response description returned from issuing institution. |
| TransAmount | | |
| CardType | 2 / alpha | Credit Card Type M = MasterCard V = Visa AX = American Express DC = Diners Card NO = Novus / Discover SE = Sears |
| Txn_number | 20 / an | Gateway Transaction identifier |
| TimedOut | true/false | Transaction failed due to a process timing out |
| Ticket | n/a | reserved |
| RecurSucess | true/false | Indicates whether the transaction successfully registered. |

18. Appendix C. Definitions of Response Fields for Vault Transactions

| Response Fields | | |
|---|--------------|---|
| Variable Name | Size/Type | Description |
| ReceiptId | 50 / an | order_id specified in request |
| reserved | N/A | Future use |
| ReponseCode | 3 / num | Transaction Response Code < 50: Transaction approved >= 50: Transaction declined |
| <u>Vault Admin Responses (i.e. res_add_cc, res_update_cc or res_delete)</u> | | |
| | 001 | Successfully registered CC details. Successfully updated CC details. Successfully deleted CC details. |
| | 983 | Can not find previous |
| | 986 | Incomplete: timed out |
| | 987 | Invalid transaction |
| | Null | Error: Malformed XML |
| TransTime | ##:##:## | Processing host time stamp (24 hour clock) |
| TransDate | yyyy-mm-dd | Processing host date stamp |
| Complete | true / false | Transaction was sent to authorization host and a response was received |
| TimedOut | true / false | Transaction failed due to a process timing out |
| data_key | 50 / an | The data_key specified in request. Or, in the case of a res_add_cc transaction it will specify the data_key created by Moneris Solutions. |
| payment_type | 3 / an | Indicates the payment type associated with the Vault profile. Possible values: cc – credit card profile |
| cust_id | 50 / an | The value stored in the vault, it will be blank if no data is stored. |
| phone | 30 / an | The value stored in the vault, it will be blank if no data is stored. |
| email | 30 / an | The value stored in the vault, it will be blank if no data is stored. |
| note | 30 / an | The value stored in the vault, it will be blank if no data is stored. |
| pan | an | The first 4 and last 4 of the credit card number. |
| exp_date | 4 / num | The credit card expiry date. |
| crypt_type | 1 / an | The E-commerce Indicator. |

19. Appendix D. Error Messages

Global Error Receipt – You are not connecting to our servers. This can be caused by a firewall or your internet connection.

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

Below are error messages that are returned in the Message field of the response.

Message: XML Parse Error in Request: <System specific detail>

Cause: For some reason an improper XML document was sent from the API to the servlet

Message: XML Parse Error in Response: <System specific detail>

Cause: For some reason an improper XML document was sent back from the servlet

Message: Transaction Not Completed Timed Out

Cause: Transaction times out before the host responds to the gateway

Message: Request was not allowed at this time

Cause: The host is disconnected

Message: Could not establish connection with the gateway:

<System specific detail>

Cause: Gateway is not accepting transactions or server does not have proper access to internet

Message: Input/Output Error: <System specific detail>

Cause: Servlet is not running

Message: The transaction was not sent to the host because of a duplicate order id

Cause: Tried to use an order id which was already in use

Message: The transaction was not sent to the host because of a duplicate order id

Cause: Expiry Date was sent in the wrong format

20. Appendix E. Example

```

<?xml version="1.0"?>
<request>
<store_id>store3</store_id>
<api_token>yesguy</api_token>
<purchase>
  <order_id>Moneris_test_June20</order_id>
  <amount>1.00</amount>
  <pan>4242424242424242</pan>
  <expdate>0901</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <cust_info>
    <billing>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </billing>
    <shipping>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </shipping>
    <email>bob@smith.com</email>
    <instructions>Please deliver to back door</instructions>
    <item>
      <name>Red Shoes</name>
      <quantity>1</quantity>
      <product_code>shoes_101</product_code>
      <extended_amount>1.00</extended_amount>
    </item>
  </cust_info>
  <recur>
    <recur_unit>month</recur_unit>
    <start_now>true</start_now>
    <start_date>2006/12/01</start_date>
    <num_recur>1</num_recur>
    <period>1</period>
    <recur_amount>1.00</recur_amount>
  </recur>
</purchase>
<refund>
  <order_id>need_order_id_from_purchasepreauth</order_id>
  <amount>2.00</amount>
  <txn_number>4678906-04-01</txn_number>
  <crypt_type>1</crypt_type>
</refund>

```



```
<preauth>
  <order_id>Moneris_test_june20_2</order_id>
  <amount>15.00</amount>
  <pan>5454545454545454</pan>
  <expdate>0702</expdate>
  <crypt_type>2</crypt_type>
  <cust_id>some_cust_id</cust_id>
  <cust_info>
    <billing>
      <first_name>Mary</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </billing>
    <shipping>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </shipping>
    <email>mary@smith.com</email>
    <instructions>Please package in wrapping paper</instructions>
    <item>
      <name>Blue Shoes</name>
      <quantity>2</quantity>
      <product_code>shoes_002</product_code>
      <extended_amount>10.00</extended_amount>
    </item>
  </cust_info>
</preauth>
<completion>
  <order_id>previous_order_id</order_id>
  <comp_amount>1.00</comp_amount>
  <txn_number>195678-03-1</txn_number>
  <crypt_type>1</crypt_type>
</completion>
<purchasecorrection>
  <order_id>some_previous_order_id</order_id>
  <txn_number>86435789-67-2</txn_number>
  <crypt_type>1</crypt_type>
</purchasecorrection>
<ind_refund>
  <order_id>need_unique_order_id</order_id>
  <amount>80.00</amount>
  <pan>4242424242424242</pan>
  <expdate>0609</expdate>
  <crypt_type>1</crypt_type>
  <cust_id>my_cust_id</cust_id>
</ind_refund>
```

```
<res_add_cc>
  <order_id>Moneris_test_Dec20_2</order_id>
  <pan>5454545454545454</pan>
  <expdate>1201</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
</res_add_cc>
<res_purchase_cc>
  <data_key>RbFbmySFugYNRSmTan6Gkm08r</data_key>
  <order_id>Moneris_test_Dec20_6</order_id>
  <cust_id>My_Customer_Name</cust_id>
  <amount>55.00</amount>
  <crypt_type>7</crypt_type>
  <cust_info>
    <billing>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </billing>
    <shipping>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </shipping>
    <email>bob@smith.com</email>
    <instructions>Please deliver to back door</instructions>
    <item>
      <name>Red Shoes</name>
      <quantity>1</quantity>
      <product_code>shoes_101</product_code>
      <extended_amount>1.00</extended_amount>
    </item>
  </cust_info>
  <recur>
    <recur_unit>month</recur_unit>
    <start_now>true</start_now>
    <start_date>2011/12/01</start_date>
    <num_recur>1</num_recur>
    <period>1</period>
    <recur_amount>1.00</recur_amount>
  </recur>
</res_purchase_cc>
<res_update_cc>
  <order_id>Moneris_test_Dec20_3</order_id>
  <data_key>XKwBrc6n8sHIJftKlmNf0TFb3</data_key>
  <expdate>1212</expdate>
</res_update_cc>
```

```
<res_update_cc>
  <order_id>Moneris_test_Dec20_4</order_id>
  <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
  <pan>5454545454545454</pan>
  <expdate>1201</expdate>
  <crypt_type>7</crypt_type>
  <cust_id>My_Customer_Name</cust_id>
  <phone>555-800-1122</phone>
  <email>my_email@mail.com</email>
  <note>I have no note at this time</note>
</res_update_cc>
<res_preauth_cc>
  <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
  <order_id>Moneris_test_Dec20_7</order_id>
  <cust_id>My_Customer_Name</cust_id>
  <amount>55.00</amount>
  <crypt_type>7</crypt_type>
  <cust_info>
    <billing>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </billing>
    <shipping>
      <first_name>Bob</first_name>
      <last_name>Smith</last_name>
      <company_name>None</company_name>
      <address>101 Main St</address>
      <city>Springfield</city>
      <province>YU</province>
      <postal_code>Z1Z1Z1</postal_code>
      <country>CAD</country>
      <phone_number>919-555-5555</phone_number>
      <fax>919-555-5550</fax>
      <tax1>0.10</tax1>
      <tax2>0.20</tax2>
      <tax3>0.00</tax3>
      <shipping_cost>0.00</shipping_cost>
    </shipping>
    <email>bob@smith.com</email>
    <instructions>Please deliver to back door</instructions>
    <item>
      <name>Red Shoes</name>
      <quantity>1</quantity>
      <product_code>shoes_101</product_code>
      <extended_amount>1.00</extended_amount>
    </item>
  </cust_info>
</res_preauth_cc>
<res_ind_refund_cc>
  <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
  <order_id>Moneris_test_Dec20_8</order_id>
  <cust_id>My_Customer_Name</cust_id>
  <amount>55.00</amount>
  <crypt_type>7</crypt_type>
</res_ind_refund_cc>
<res_delete>
  <order_id>Moneris_test_Dec20_5</order_id>
  <data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
</res_delete>
</request>
```