eSelect PLUS

Merchant Integration Guide

XML File Upload – v1.03

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

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 – (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.

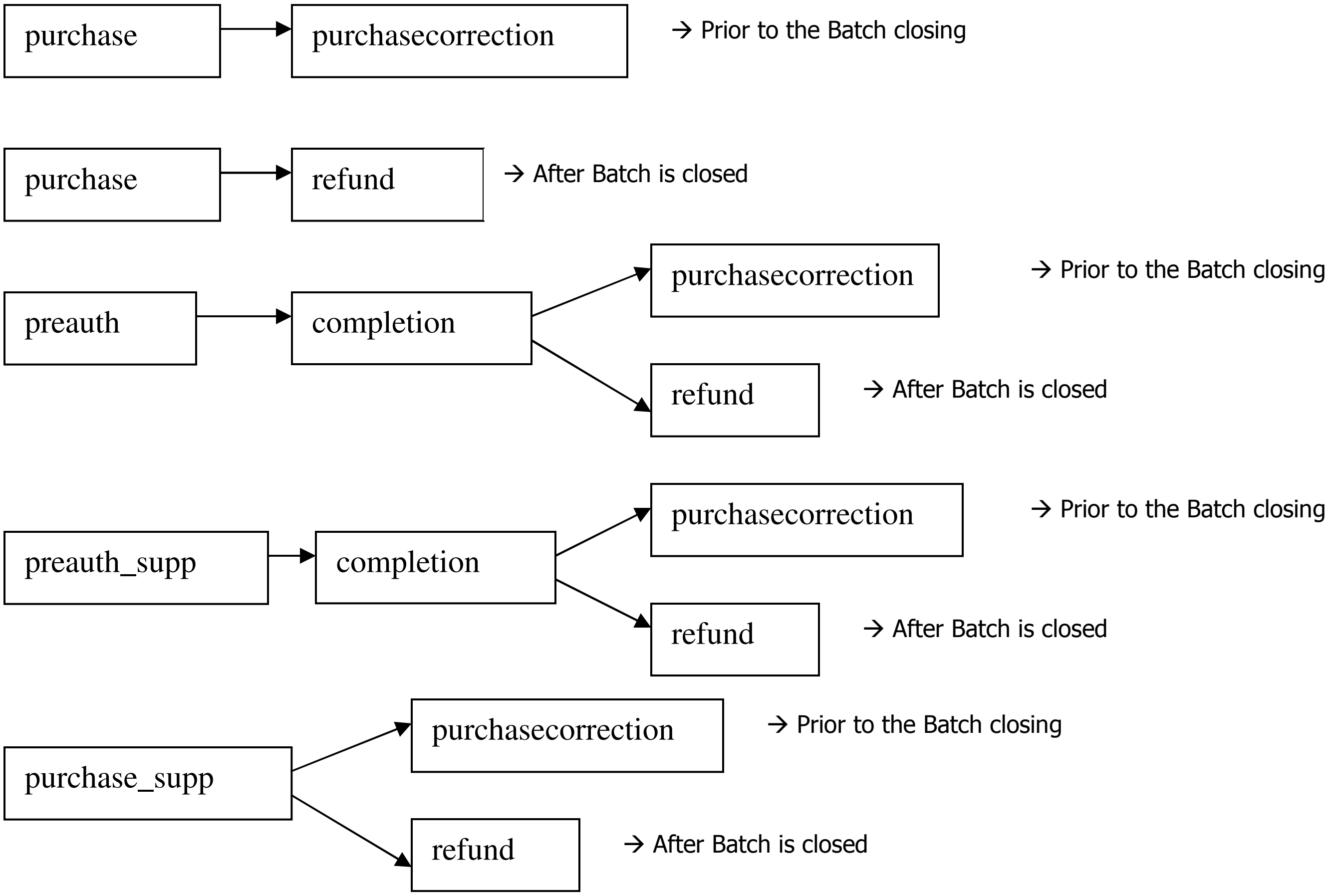
Void – (Correction / Purchase Correction) Purchases and Captures can be voided the same day* that they occur. A Void must be for the full amount of the transaction and will remove any record of it from the cardholder's statement.

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.

* 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

Mag Swipe Transactions

Mag Swipe Purchase – (sale) The Mag Swipe Purchase transaction requires a credit card to be swiped to collect the track2 data. 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 to collect the track2 data. 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.

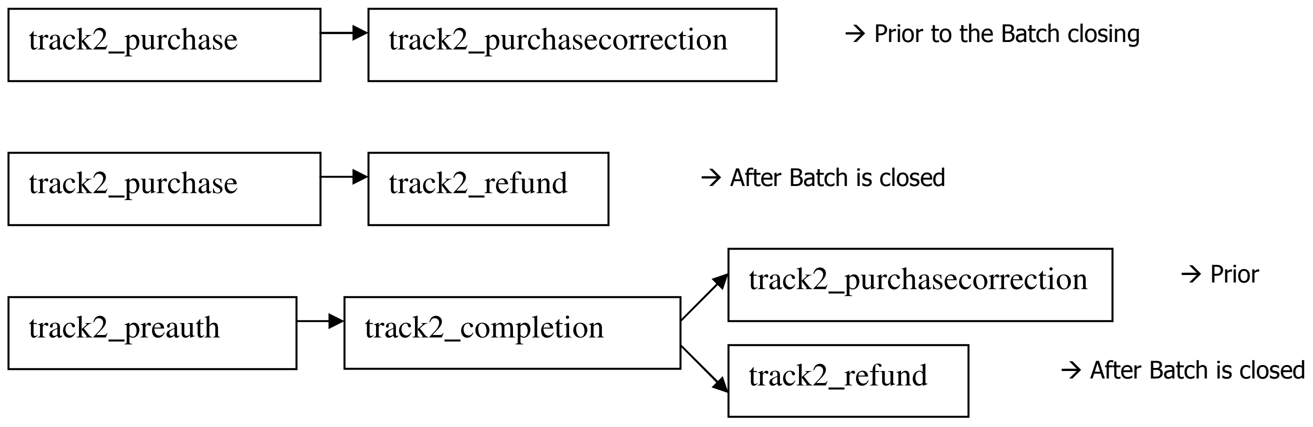
Mag Swipe Void – (Correction / Purchase Correction) Mag Swipe Purchases and Mag Swipe Captures can be voided the same day* that they occur. A Mag Swipe Void must be for the full amount of the transaction and will remove any record of it from the cardholder's statement.

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.

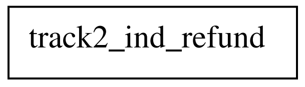
Mag Swipe Independent Refund – (Credit) A Mag Swipe Independent Refund requires a credit card to be swiped to collect the track2 data. It can be performed to credit money to this particular credit card. This transaction does not require a prior Mag Swipe Purchase or Mag Swipe Capture.

* 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



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, both basic transactions and Mag Swipe transactions. Mag Swipe transactions allow the merchant to submit track2 details that have been collected by swiping a credit card through a card reader. 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 'expdate' variables. If all three fields are submitted, the track2 details will be used to process the transaction. For a full description of each field please refer to Appendix A. 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 D.

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
	Mag Swipe Transaction Types
Mag Swipe Purchase	track2_purchase, order_id, cust_id, amount, track2, pan, expdate, pos_code
Mag Swipe Pre-authorization	track2_preauth, order_id, cust_id, amount, track2, pan, expdate, pos_code
Mag Swipe Capture	track2_completion, order_id (from track2_preauth), comp_amount, txn_number (from track2_preauth)
Mag Swipe Void	track2_purchasecorrection, order_id (from orig txn), txn_number (from track2_purchase or track2_completion)
Mag Swipe Refund	track2_refund, order_id (from original txn), amount, txn_number (from track2_purchase or track2_completion)
Mag Swipe Independent Refund	track2_ind_refund, order_id, cust_id, amount, track2, pan, expdate, pos_code

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 Appendix D.

Optional Fields	
Transaction Type	**Fields**
Basic Purchase	cust_info – all fields contained within cust_info are required recur - all fields contained within recur are required
Basic Pre-authorization	cust_info – all fields contained within cust_info are required

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>

<track2_purchase>
        <order_id>track2_testing_purch_1</order_id>
        <cust_id>my cust id</cust_id>
        <amount>1.00</amount>
        <track2>;5258984987184986=06061016091001060602?</track2>
        <pan></pan>
        <expdate></expdate>
        <pos_code>00</pos_code>
</track2_purchase>

<track2_preauth>
        <order_id>track2_testing_preauth_1</order_id>
        <cust_id>my cust id</cust_id>
        <amount>25.00</amount>
        <track2>;5258984987184986=06061016091001060602?</track2>
        <pan></pan>
        <expdate></expdate>
        <pos_code>00</pos_code>
</track2_preauth>
</request>
```

NOTE

In a mag swipe/track2 transaction, you must still include all of the required tags such as 'pan' and 'expdate'. This may be seen in examples 3 and 4 above where the 'pan' and 'exp_date' have been left blank, but the XML tags have still been included.

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>
        <ReceiptId>track2_testing_purch_1</ReceiptId>
        <ReferenceNum>660021810013842370</ReferenceNum>
        <ResponseCode>027</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>007427</AuthCode>
        <TransTime>15:00:36</TransTime>
        <TransDate>2006-11-24</TransDate>
        <TransType>00</TransType>
        <Complete>true</Complete>
        <Message>APPROVED           *                    =</Message>
        <TransAmount>1.00</TransAmount>
        <CardType>M</CardType>
        <TransID>97572-237-0</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
</receipt>
```

```
<receipt>
        <ReceiptId>track2_testing_preauth_1</ReceiptId>
        <ReferenceNum>660021810013842380</ReferenceNum>
        <ResponseCode>027</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>009429</AuthCode>
        <TransTime>15:00:36</TransTime>
        <TransDate>2006-11-24</TransDate>
        <TransType>01</TransType>
        <Complete>true</Complete>
        <Message>APPROVED           *                    =</Message>
        <TransAmount>25.00</TransAmount>
        <CardType>M</CardType>
        <TransID>97573-238-0</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
</receipt>
</response>
```

7. 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 (trans_type)		All transactions with extra customer and order details must be sent with trans_type as 'purchase' or 'preauth'. 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.
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
pan	20 / num	Credit Card number – no spaces, no dashes (numeric only)
expdate	4 / num	**Format – YYMM. Please note, this is opposite from what appears on the card.**
crypt_type	1 / num	E-Commerce Indicator: 1 - Mail Order/Telephone Order - Recurring Please refer to Appendix A 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 quanitity > 0 or the item will not be added to the item list (ie. 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>
                        <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>

</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>
</response>
```

8. 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 (trans_type)		All recurs must be sent with trans_type as purchase. 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.
pan	20 / num	Credit Card number – no spaces, no dashes (numeric only)
expdate	4 / num	**Format – YYMM** **Please note, this is opposite from what appears on the card.**
crypt_type	1 / num	E-Commerce Indicator: 2 - Mail Order/Telephone Order - Recurring Please refer to appendix A 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</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>2006/10/30</start_date>
            <num_recurs>4</num_recurs>
            <period>2</period>
            <recur_amount>1.00</recur_amount>
        </recur>
    </purchase>
    <purchase>
        <order_id>464571415532</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>2006/09/15</start_date>
            <num_recurs>10</num_recurs>
            <period>30</period>
            <recur_amount>1.00</recur_amount>
        </recur>
    </purchase>
    <purchase>
        <order_id>464571415533</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>2006/11/11</start_date>
            <num_recurs>12</num_recurs>
            <period>1</period>
            <recur_amount>1.00</recur_amount>
        </recur>
     </purchase>
</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.

Corresponding Example Response

```
<?xml version="1.0"?>
<response>
<receipt>
        <ReceiptId>464571415531</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</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</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>
</response>
```

9. 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)**

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.

10. 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. 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://www3.moneris.com/connect/en/documents/index.html and download the Response Code document.

11. 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

12. 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 9.

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. For track2, you may also test by swiping your own card through a mag swipe reader and collecting the track2 data.

Test Card Numbers	
Card Plan	Card Number
MasterCard	5454545454545454
Visa	4242424242424242
Amex	373599005095005
Diners	36462462742008
Track2	;5258984987184986=06061016091001060602?

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 store 3, 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://www3.moneris.com/connect/en/documents/index.html.

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

13. 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.

14. 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.**

15. Appendix A. Definitions of Required Fields

Required Fields		
Variable Name	**Size/Type**	**Description**
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: • purchase • preauth • completion • purchasecorrection • refund • ind_refund • track2_purchase • track2_preauth • track2_completion • track2_purchasecorrection • track2_refund • track2_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 MMYY
track2		This is a string that is retrieved from the mag swipe of a credit card by swiping the credit card through a card reader. It is part of a mag swipe/track2 transaction.
amount	9 / decimal	Amount of the transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
crypt_type	1 / an	E-Commerce Indicator: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 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
pos_code	2 / num	Under normal presentment situations the value should be ‘00’. If the solution is not “merchant and cardholder present” please call the support desk and we will provide the proper POS Code.
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.

16. Appendix B. Definitions of Response Fields

Response Fields		
Variable Name	**Size/Type**	**Description**
order_id	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. Moneris Host Transaction identifier.
ReponseCode	3 / num	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://www3.moneris.com/connect/en/documents/index.html
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
Complete	True/False	Transaction was sent to authorization host and a response was received
Message	100 / an	Response description returned from issuing institution.
TransAmount		
CardType	2 / alpha	Credit Card Type
Txn_number	20 / an	Gateway Transaction identifier
TimedOut	True/False	Transaction failed due to a process timing out
Ticket	n/a	reserved

17. Appendix C. 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

18. Appendix D. 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_recurs>1</num_recurs>
                <period>1</period>
                <recur_amount>1.00</recur_amount>
        </recur>
</purchase>
```

```
<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>shooes_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>

<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>
```

```
<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>



<track2_purchase>
        <order_id>track2_testing_purch_1</order_id>
        <cust_id>my cust id</cust_id>
        <amount>1.00</amount>
        <track2>;5258984987184986=06061016091001060602?</track2>
        <pan></pan>
        <expdate></expdate>
        <pos_code>00</pos_code>
</track2_purchase>



<track2_preauth>
        <order_id>track2_testing_preauth_1</order_id>
        <cust_id>my cust id</cust_id>
        <amount>25.00</amount>
        <track2>;5258984987184986=06061016091001060602?</track2>
        <pan></pan>
        <expdate></expdate>
        <pos_code>00</pos_code>
</track2_preauth>



<track2_completion>
        <order_id>track2testing2</order_id>
        <comp_amount>1.00</comp_amount>
        <txn_number>97548-218-0</txn_number>
        <pos_code>00</pos_code>
</track2_completion>



<track2_purchasecorrection>
        <order_id>track2testing1</order_id>
        <txn_number>97547-217-0</txn_number>
</track2_purchasecorrection>



<track2_refund>
        <order_id>track2_testing_purch_2</order_id>
        <amount>1.00</amount>
        <txn_number>97571-236-0</txn_number>
</track2_refund>



<track2_ind_refund>
        <order_id>track2_testing_indrefund_1</order_id>
        <cust_id>my cust id</cust_id>
        <amount>2.00</amount>
        <track2>;5258984987184986=06061016091001060602?</track2>
        <pan>5454545454545445</pan>
        <expdate>0909</expdate>
        <pos_code>00</pos_code>
</track2_ind_refund>

</request>
```