



Merchant Integration Guide

Transaction DTD MSR (Canada)

v.1.0.0 Draft



Revision Number	Date	Description
V1.0.0	February 27, 2013	-Created document, added ROAMpay reader XML format

Table of Contents

1.	About this Documentation	5
2.	What is the Process I will need to follow?	5
3.	Transaction Types and Transaction Flow	6
4.	What Information do I need to include in a Transaction Request.	8
5.	Request DTD	9
6.	Response DTD	12
7.	What Information will I get as a Response to My Transaction Request?	13
8.	How Do I Test My Solution?	13
9.	What Do I Need to Include in the Receipt?	14
10.	How Do I Activate My Store?	15
11.	How Do I Configure My Store For Production?	15
12.	How Do I Get Help?	15
13.	Appendix A. Definition of Request Fields	16
14.	Appendix B. Definitions of Response Fields	18
15.	Appendix C. CustInfo Fields	19
16.	Appendix D. Recur Fields	20
17.	Appendix E. Error Messages	23
18.	Appendix F. Card Validation Digits (CVD)	24
19.	Appendix G. Address Verification Service (AVS)	25
20.	Appendix H. Additional Information for CVD and AVS	27
21.	Appendix I. Sample Receipt	28

****** PLEASE READ CAREFULLY******

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

1. About this Documentation

The eSELECTplus payment gateway supports credit card transactions in XML format over the HTTPS protocol. This document contains detailed information on the request and response transaction requirements of eSELECTplus' XML format. When creating custom API's, these requirements must be met in order for transactions to be sent to eSELECTplus in the proper format.

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

You will need to follow these steps.

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

Note:

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

As a Moneris Solutions client or partner using this method of integration, your solution must demonstrate compliance to the Payment Card Industry Data Security Standard (PCI DSS) and/or the Payment Application Data Security Standard (PA DSS). These standards are designed to help the cardholders and merchants in such ways as they ensure credit card numbers are encrypted when transmitted/stored in a database and that merchants have strong access control measures.

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

For more information on how to get your application PCI-DSS compliant, please contact our Integration Specialists and visit <http://www.eselectplus.ca/en/downloadable-content> to download the PCI-DSS Implementation Guide.

3. Transaction Types and Transaction Flow

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

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.

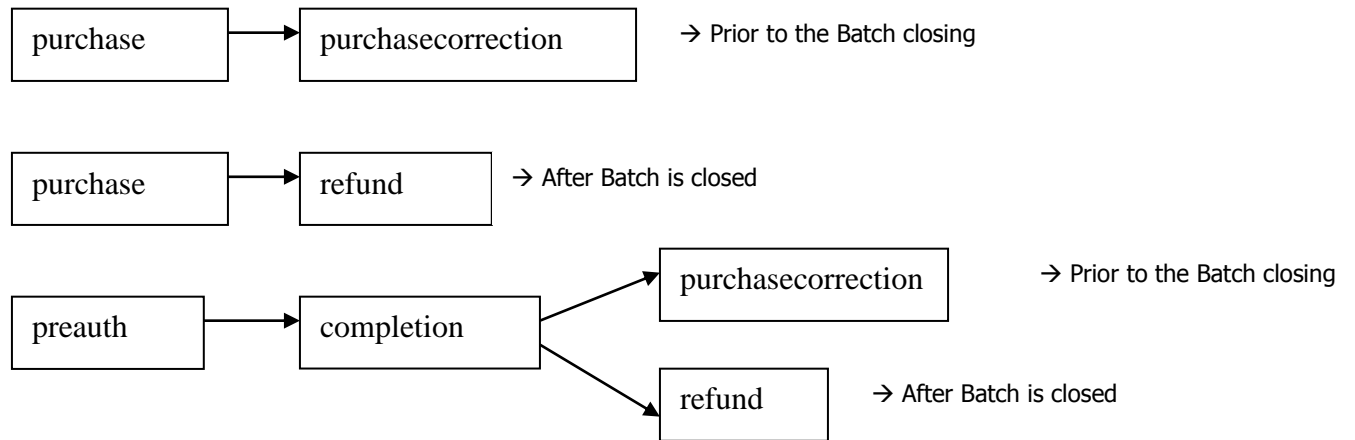
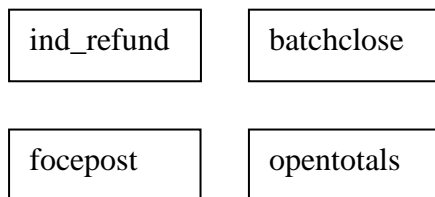
Capture – (Completion / PreAuth Completion) Once a PreAuth is obtained the funds that are locked need to be retrieved from the customer's credit card. The Capture retrieves the locked funds and readies them for settlement into the merchant's account.

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.

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

* A Void can be performed against a transaction as long as the batch that contains the original transaction remains open. When using the automated closing feature Batch Close occurs daily between 10 – 11 pm EST.

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

4. 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. Definition of Request Fields. Please note that the fields must be in the correct order.

Required Fields	
Transaction Type	Fields
<u>Basic Transaction Types</u>	
Purchase	purchase (order_id, cust_id (optional), amount, pan, expdate, crypt_type, cust_info (optional), avs_info (optional), cvd_info (optional), recur (optional))
Pre-authorization	preauth (order_id, cust_id (optional), amount, pan, expdate, crypt_type, cust_info (optional), avs_info (optional), cvd_info (optional))
Capture	completion (order_id, comp_amount, txn_number, crypt_type, ship_indicator (optional))
Void	purchasecorrection (order_id, txn_number, crypt_type, ship_indicator(optional))
Refund	refund (order_id, amount, txn_number, crypt_type)
Independent Refund	ind_refund (order_id, cust_id (optional), amount, pan, expdate, crypt_type)
<u>Administrative Transactions</u>	
Force post	forcepost (order_id, cust_id (optional), amount, pan, expdate, crypt_type, auth_code)
Recur Update	recur_update (order_id, cust_id (optional), pan (optional), expdate (optional), recur_amount (optional), add_num_recur (optional), total_num_recur (optional), hold (optional), terminate (optional))
Batch Close	batchclose (ecr_number)
Open Total	opentotals (ecr_number)
<u>INTERAC Online Transactions</u>	
INTERAC Online Purchase	idebit_purchase(order_id, cust_id (optional), amount, idebit_track2)
INTERAC Online Refund	idebit_refund(order_id, amount, txn_number)
<u>VbV Transactions</u>	
Verified by Visa Purchase	cavv_purchase (order_id, cust_id (optional), amount, pan, expdate, cavv, cust_info (optional), avs_info (optional), cvd_info (optional), recur (optional))
Verified by Visa Preauth	cavv_preauth (order_id, cust_id (optional), amount, pan, expdate, cavv, cust_info (optional), avs_info (optional), cvd_info (optional))

* Please refer to Appendix A. Definition of Request Fields for field definitions.

The Transaction Types (ex. purchase or ind_refund) is case sensitive and must be all in lower case.



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.

5. Request DTD

<!-- The Request DTD -->

<!-- Main Elements -->

```
<!ELEMENT request (store_id, api_token, (purchase | refund | ind_refund | preauth | completion |
purchasecorrection | forcepost | idebit_purchase | idebit_refund | cavv_preauth | cavv_purchase | batchclose |
opentotals | recur_update | track2_purchase | track2_refund | track2_ind_refund | track2_preauth |
track2_completion | track2_purchasecorrection))>
<!ELEMENT store_id (#PCDATA)>
<!ELEMENT api_token (#PCDATA)>

<!ELEMENT track2_preauth (order_id, cust_id?, amount, pan, expdate, track2, pos_code, cust_info? )>
<!ELEMENT track2_purchase (order_id, cust_id?, amount, pan, expdate, track2, pos_code, cust_info? )>
<!ELEMENT track2_refund (order_id, amount, txn_number)>
<!ELEMENT track2_ind_refund (order_id, cust_id?, amount, pan, expdate, track2, pos_code)>
<!ELEMENT track2_completion (order_id, comp_amount, txn_number)>
<!ELEMENT track2_purchasecorrection (order_id, txn_number)>
<!ELEMENT enc_track1_preauth (order_id, cust_id?, amount, pos_code, device_type, cust_info?, rp_enc_track?,
rp_format_id?, rp_last4_digits?, rp_cardholder_name?, rp_expdate?, rp_partial_track?, rp_ksn? )>
<!ELEMENT enc_track1_purchase (order_id, cust_id?, amount, pos_code, device_type, cust_info?, rp_enc_track?,
rp_format_id?, rp_last4_digits?, rp_cardholder_name?, rp_expdate?, rp_partial_track?, rp_ksn? )>
<!ELEMENT enc_track1_ind_refund (order_id, cust_id?, amount, pos_code, device_type, cust_info?,
rp_enc_track?, rp_format_id?, rp_last4_digits?, rp_cardholder_name?, rp_expdate?, rp_partial_track?, rp_ksn? )>
<!ELEMENT purchase (order_id, cust_id?, amount, pan, expdate, crypt_type, cust_info?, avs_info?, cvd_info?,
recur? )>
<!ELEMENT refund (order_id, amount, txn_number, crypt_type)>
<!ELEMENT ind_refund (order_id, cust_id?, amount, pan, expdate, crypt_type)>
<!ELEMENT preauth (order_id, cust_id?, amount, pan, expdate, crypt_type, cust_info?, avs_info?, cvd_info?)>
<!ELEMENT completion (order_id, comp_amount, txn_number, crypt_type)>
<!ELEMENT purchasecorrection (order_id, txn_number, crypt_type)>
<!ELEMENT forcepost (order_id, cust_id?, amount, pan, expdate, crypt_type, auth_code)>
<!ELEMENT idebit_purchase (order_id, cust_id?, amount, idebit_track2) >
<!ELEMENT idebit_refund (order_id, amount, txn_number)>
<!ELEMENT cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv, cust_info?, avs_info?, cvd_info?,
recur?)>
<!ELEMENT cavv_preauth (order_id, cust_id?, amount, pan, expdate, cavv, cust_info?, avs_info?, cvd_info?)>
<!ELEMENT recur_update (order_id, cust_id?, pan?, expdate?, recur_amount?, add_num_recur?,
total_num_recur?, hold?, terminate?)>
<!ELEMENT batchclose (ecr_number)>
<!ELEMENT opentotals (ecr_number)>
<!ELEMENT order_id (#PCDATA)>
<!ELEMENT cust_id (#PCDATA)>
<!ELEMENT track2 (#PCDATA)>
<!ELEMENT pos_code (#PCDATA)>
<!ELEMENT device_type (#PCDATA)>
<!ELEMENT txn_number (#PCDATA)>
```

```
<!ELEMENT crypt_type (#PCDATA)>
<!ELEMENT auth_code (#PCDATA)>
<!ELEMENT cavv (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT comp_amount (#PCDATA)>
<!ELEMENT pan (#PCDATA)>
<!ELEMENT idebit_track2 (#PCDATA)>
<!ELEMENT expdate (#PCDATA)>
<!ELEMENT ecr_number (#PCDATA)>
<!ELEMENT add_num_rekurs (#PCDATA)>
<!ELEMENT total_num_rekurs (#PCDATA)>
<!ELEMENT hold (#PCDATA)>
<!ELEMENT terminate (#PCDATA)>
<!ELEMENT rp_enc_track (#PCDATA)>
<!ELEMENT rp_format_id (#PCDATA)>
<!ELEMENT rp_last4_digits (#PCDATA)>
<!ELEMENT rp_cardholder_name (#PCDATA)>
<!ELEMENT rp_expdate (#PCDATA)>
<!ELEMENT rp_partial_track (#PCDATA)>
<!ELEMENT rp_ksn (#PCDATA)>
```

<!-- start Cust Info -->

```
<!ELEMENT cust_info (billing, shipping, email, instructions, item+)>
<!ELEMENT billing (first_name, last_name, company_name, address, city, province, postal_code, country,
phone_number, fax, tax1, tax2, tax3, shipping_cost)>
<!ELEMENT shipping (first_name, last_name, company_name, address, city, province, postal_code, country,
phone_number, fax, tax1, tax2, tax3, shipping_cost)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT instructions (#PCDATA)>
<!ELEMENT item (name, quantity, product_code, extended_amount)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT company_name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT province (#PCDATA)>
<!ELEMENT postal_code (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
<!ELEMENT product_code (#PCDATA)>
<!ELEMENT extended_amount (#PCDATA)>
<!ELEMENT tax1 (#PCDATA)>
<!ELEMENT tax2 (#PCDATA)>
<!ELEMENT tax3 (#PCDATA)>
<!ELEMENT shipping_cost (#PCDATA)>
```

<!-- start AVS -->

```
<!ELEMENT avs_info (avs_street_number, avs_street_name, avs_zipcode)>
<!ELEMENT avs_street_number (#PCDATA)>
<!ELEMENT avs_street_name (#PCDATA)>
<!ELEMENT avs_zipcode (#PCDATA)>
```

<!-- start CVD -->

```
<!ELEMENT cvd_info (cvd_indicator, cvd_value)>
```

```
<!ELEMENT cvd_indicator (#PCDATA)>  
<!ELEMENT cvd_value (#PCDATA)>
```

<!-- start Recur -->

```
<!ELEMENT recur (recur_unit, start_now, start_date, num_recur, period, recur_amount)>  
<!ELEMENT recur_unit (#PCDATA)>  
<!ELEMENT start_now (#PCDATA)>  
<!ELEMENT start_date (#PCDATA)>  
<!ELEMENT num_recur (#PCDATA)>  
<!ELEMENT period (#PCDATA)>  
<!ELEMENT recur_amount (#PCDATA)>
```

<-- start RecurUpdate -->

```
<!ELEMENT recur_update (order_id?, cust_id?, pan?, expdate?, recur_amount?, add_num_recur?,  
total_num_recur?, hold?, terminate?)>  
<!ELEMENT order_id (#PCDATA)>  
<!ELEMENT cust_id (#PCDATA)>  
<!ELEMENT pan (#PCDATA)>  
<!ELEMENT expdate (#PCDATA)>  
<!ELEMENT recur_amount (#PCDATA)>  
<!ELEMENT add_num_recur (#PCDATA)>  
<!ELEMENT total_num_recur (#PCDATA)>  
<!ELEMENT hold (#PCDATA)>  
<!ELEMENT terminate (#PCDATA)>
```

6. Response DTD

<!-- The Response DTD -->

<!-- Main Elements -->

<!ELEMENT response receipt>

<!ELEMENT receipt (ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate, TransType, Complete, Message, TransAmount, CardType, TransID, TimedOut, BankTotals, Ticket, RecurSuccess?, CvdResultCode?, AvsResultCode?, RecurUpdateSuccess?, NextRecurDate?, RecurEndDate?)>

<!ELEMENT ReceiptId (#PCDATA)>
<!ELEMENT ReferenceNum (#PCDATA)>
<!ELEMENT ResponseCode (#PCDATA)>
<!ELEMENT ISO (#PCDATA)>
<!ELEMENT AuthCode (#PCDATA)>
<!ELEMENT TransTime (#PCDATA)>
<!ELEMENT TransDate (#PCDATA)>
<!ELEMENT TransType (#PCDATA)>
<!ELEMENT Complete (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
<!ELEMENT TransAmount (#PCDATA)>
<!ELEMENT CardType (#PCDATA)>
<!ELEMENT TransID (#PCDATA)>
<!ELEMENT TimedOut (#PCDATA)>
<!ELEMENT BankTotals (ECR)>
<!ELEMENT Ticket (#PCDATA)>
<!ELEMENT CvdResultCode (ECR)>
<!ELEMENT AvsResultCode (#PCDATA)>
<!ELEMENT RecurSuccess (#PCDATA)>
<!ELEMENT RecurUpdateSuccess (#PCDATA)>
<!ELEMENT NextRecurDate (#PCDATA)>
<!ELEMENT RecurEndDate (#PCDATA)>

<!ELEMENT ECR (term_id, closed, Card+)>

<!ELEMENT Card (CardType, (Purchase | Refund | Correction)+)>

<!ELEMENT term_id (#PCDATA)>
<!ELEMENT closed (#PCDATA)>

<!ELEMENT Purchase (Count, Amount)>
<!ELEMENT Refund (Count, Amount)>
<!ELEMENT Correction (Count, Amount)>

<!ELEMENT Count (#PCDATA)>
<!ELEMENT Amount (#PCDATA)>

7. 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

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 <http://www.eselectplus.ca/en/downloadable-content> and download the Response Code document.

8. How Do I Test My Solution?

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

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

Test IDs			
store_id	api_token	Username	Password
store1	yesguy	DemoUser	password
store2	yesguy	DemoUser	password
store3	yesguy	DemoUser	password
store5 *	yesguy	DemoUser	password
moneris **	hurgle	DemoUser	password

* “store5” is for testing eFraud (AVS & CVD)

** the “moneris” store is for testing VBV and SecureCode

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

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

To access the Merchant Resource Centre in the test environment go to <https://esqa.moneris.com/mpg> and use the logins provided in the Test ID table.

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

The test environment will approve and decline credit card transactions based on the penny value of the amount field. For example, a transaction made for the amount of \$9.00 or \$1.00 will approve since the .00 penny value is set to approve in the test environment. Transactions in the test environment should not exceed \$11.00. This limit does not exist in the production environment. For a list of all current test environment responses for various penny values, please see the Test Environment Penny Response table as well as the Test Environment eFraud Response table, available at <http://www.eselectplus.ca/en/downloadable-content>.



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

9. What Do I Need to Include in the Receipt?

Visa and MasterCard expect certain variables be returned to the cardholder and presented as a receipt when a transaction is approved. These 12 fields are listed below. A sample receipt is provided in Appendix I. Sample Receipt.

1. Amount
2. Transaction Type
3. Date and Time
4. Auth Code
5. Response Code
6. ISO Code
7. Response Message
8. Reference Number
9. Goods and Services Order
10. Merchant Name
11. Merchant URL
12. Cardholder Name
13. Return Policy (only a requirement for e-commerce transactions)

10. 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 the Store ID and API Token to send transactions through the API.

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 the APIs that you are using. This will allow us to keep you up to date regarding any changes to the APIs that may affect your store. Also, this is where you may locate your production API Token.

11. How Do I Configure My Store For Production?

Once you have completed your testing you are ready to point your store to the production host. You will need to change the "host" to be www3.moneris.com. You will also need to change the store_id to reflect your production store ID and well the api_token must be changed to your production token to reflect the token that you received during activation.

Once you are in production you will access the Merchant Resource Centre at <https://www3.moneris.com/mpg>. You can use the store administrator ID you created during the activation process and then create additional users as needed.

For further information on how to use the Merchant Resource Centre please see the eSELECTplus Merchant Resource Centre User's Guide which is available at <http://www.eselectplus.ca/en/downloadable-content>.

12. How Do I Get Help?

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

For Technical Support (7/24):
Phone: 1-866-319-7450 (Technical Difficulties)

For Integration Support (M-F 8am-8pm EST):
Phone: 1-866-562-4354
Email: eselectplus@moneris.com

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

13. Appendix A. Definition of Request 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: <ul style="list-style-type: none"> • purchase • refund • ind_refund • preauth • completion • purchasecorrection • forcepost • idebit_purchase • idebit_refund • cavv_preauth • cavv_purchase • batchclose • opentotals
order_id	50 / an	Merchant defined unique transaction identifier - must be unique for every Purchase, Preauth and Independent Refund attempt. For Refunds, completion and purchasecorrection 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
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
ship_indicator	1 / an	P – Partial shipment, F – Final shipment (case sensitive must be upper case)
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.
cavv		This is a value that is provided by the Moneris MPI or by a third party MPI. It is part of a VBV transaction.

avs_street_number		Street Number & Street Name (max – 19 digit limit for street number and street name combined). This must match the address that the issuing bank has on file.
avs_street_name	19 / an	
avs_zipcode	10 / an	Zip or Postal Code – This must match what the issuing banks has on file.
cvd_value	4 / num	Credit Card CVD value – this number accommodates either 3 or 4 digit CVD values. Note: The CVD value supplied by the cardholder should simply be passed to the eSelectPlus payment gateway. Under no circumstances should it be stored for subsequent uses or displayed as part of the receipt information.
cvd_indiator	1 / num	CVD presence indicator (1 digit – refer to Appendix F. Card Validation Digits (CVD))
auth_code	8 / an	Use when performing a Forcepost transaction. The auth code is a number given by the issuer which refers to the particular transaction previously authorized on an IVR or equivalent terminal. To obtain an Authorization Code for Visa and MasterCard, please call 1-866-802-2637. To obtain an Authorization Code for AmEx, please contact American Express directly.
comp_amount	9 / decimal	Used when performing a Capture transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
ecr_number	8 / an	Terminal ID – no spaces or dashes. The ECR/terminal number is 8 digits long and generally begins with the number “66” (ex. 66012345). The terminal ID is assigned when the merchant account is first set up. This number will be returned in the response as part of the ReferenceNum.
idebit_track2	37 / an	This value will be returned by the issuer. It includes the PAN, expiry date, and transaction ID. Characters allowed are ISO-8859-1, restricted to single-byte codes, hex 20 to 7E (consistent with US-ASCII and ISO-8859-1 Latin-1). As part of the validation process, a MOD 10 check will have to be performed on the PAN portion (i.e., all characters before the '=' sign) of the track2 value.



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

NOTE

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

14. 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 http://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
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 / an	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
RecurSucess	true / false	Indicates whether the transaction successfully registered.
AvsResultCode	1 / an	Indicates the Address Verification Service (AVS) result. Refer to Appendix G. Address Verification Service (AVS).
CvdResultCode	2 / an	Indicates the Card Validation Digits (CVD) result. Refer to Appendix F. Card Validation Digits (CVD).

15. Appendix C. CustInfo Fields

Field Definitions		
Field Name	Size/Type	Description
Billing and Shipping Information		
NOTE: The fields for billing and shipping information are identical.		
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: You may send multiple items		
description	30 / an	
quantity	10 / num	You must send a quantity > 0 or the item will not be added to the item list (ie. minimum 1, maximum 9999999999)
product_code	30 / 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	

If you send characters that are not included in the allowed list, these extra transaction details may not be stored.


NOTE

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

Also, the data sent in Billing and Shipping Address fields will not be used for any address verification. Please refer to the section Appendix G. Address Verification Service (AVS) for further details about Address Verification Service (AVS).

16. Appendix D. Recur Fields

Recur Request Fields		
Variable Name	Size/Type	Description
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.
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.
num_rekurs	1 – 99 / num	The number of times to recur the transaction. Moneris advises not set the duration longer than 5 - 10 years in the future.
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.
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=week -> 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.
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.
amount	9 / decimal	When start_now is set to true the amount field in the transaction array becomes the amount to be billed immediately. When start_now is set to false the amount field in the transaction array should be the same as the recur_amount field.

Recur Request Examples	
Recur Request Example	Description
recur_unit='month', start_date='2006/07/02', num_rekurs='12', start_now='false', period = '2', recur_amount= '30.00'	In the example to the left the first transaction will occur in the future on July 2 nd 2006. It will bill \$30.00 every 2 months on the 2 nd of each month. The card will be billed a total of 12 times.
recur_unit='week', start_date='2006/07/02', num_rekurs='26', start_now='true', period = '2', recur_amount= '30.00' amount='15.00'	In the example on the left the first charge will be billed immediately. The initial charge will be for \$15.00. Then starting on July 2 nd 2006 the credit card will be billed \$30.00 every 2 weeks for 26 recurring charges. The card will be billed a total of 27 times. (1 x \$15.00 (immediate) and 26 x \$30.00 (recurring))


NOTE

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

Recur Update Request Fields		
Variable Name	Size/Type	Description
cust_id	50 / an	This updates the current cust_id associated with the recurring transaction and will be submitted with all future recurring purchases.
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. This will be the new credit card number charged with all future recurs. This field pertains only to credit card transactions.
expdate	YYMM / num	Expiry Date - format YYMM no spaces or slashes, replaces the current expiry date in the payment details and must be today's date or later. PLEASE NOTE THAT THIS IS REVERSED FROM THE DATE DISPLAYED ON THE PHYSICAL CARD WHICH IS MMY
avs_street_number	19 / an	Street Number & Street Name (max – 19 digit limit for street number and street name combined). This must match the address that the issuing bank has on file. The updated AVS details will be submitted for all future credit card recurs. Please note; the store must have the AVS feature enabled.
avs_street_name		
avs_zipcode	9 / an	Zip or Postal Code – This must match what the issuing banks has on file.
recur_amount	9 / decimal	Amount of all future recurring transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99.
add_num	1-999 / num	This is the number of recurring transactions to be added to the current total number of recurs on file. Example: num_recurs* = 5, add_num = 2, New total number of recurs = 7 *the "num_recurs" initially sent in while registering the recurring transaction. Please refer to Recur Request Fields table for variable definition.
total_num	1-999 / num	This is an update to replace the current total number of recurs on file. Example: num_recurs* = 5, total_num = 2, New total number of recurs = 2 *the "num_recurs" initially sent in while registering the recurring transaction. Please refer to Recur Request Fields table for variable definition.
hold	true / false	A transaction can be put 'On Hold' at any time. While a transaction is 'On Hold' it will not be billed when the time comes for it to recur, but the number of recurs will be decremented.
terminate	true / false	A Recurring Billing transaction can be Terminated at any time. A terminated Recurring transaction can no longer be reactivated.

Recur Update Response codes:

The Recur Update response is a 3 digit numeric value. The following is a list of all possible responses once a Recur Update transaction has been sent thru.

Recur Update RESPONSE CODES	
RESULT VALUE	DEFINITION
001	Recurring transaction successfully updated (optional: terminated)
983	Can not find the previous transaction
984	Data error: (optional: field name)
985	Invalid number of recurs
986	Incomplete: timed out
null	Error: Malformed XML

**NOTE**

When completing the update recurring billing portion please keep in mind that the recur bill dates cannot be changed to have an end date greater than 10 years from today and cannot be changed to have an end date end today or earlier.

17. Appendix E. 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 F. Card Validation Digits (CVD)

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

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

CVD Indicator:

0 = CVD value is deliberately bypassed or is not provided by the merchant.

1 = CVD value is present.

2 = CVD value is on the card, but is illegible.

9 = Cardholder states that the card has no CVD imprint.

CVD Response:

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

CVD Response Code:

M = Match

Y = Match for AmEx/JCB

N = No Match

P = Not Processed

S = CVD should be on the card, but Merchant has indicated that CVD is not present

U = Issuer is not a CVD participant

D = Invalid security code for AmEx/JCB



NOTE

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

***For additional information on how to handle these responses, please refer to Appendix H. Additional Information for CVD and AVS.**

19. Appendix G. Address Verification Service (AVS)

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

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

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

Table of possible Visa and MC AVS result codes			
VALUE	DESCRIPTION	Domestic	International
A	Street addresses match. The street addresses match but the postal/ZIP codes do not, or the request does not include the postal/ZIP code.	√	√
B	Street addresses match. Postal code not verified due to incompatible formats. (Acquirer sent both street address and postal code.)	√	√
C	Street address and postal code not verified due to incompatible formats. (Acquirer sent both street address and postal code.)	√	√
D	Street addresses and postal codes match.		√
G	Address information not verified for international transaction.		√
I	Address information not verified.		√
M	Street address and postal code match.		√
N	No match. Acquirer sent postal/ZIP code only, or street address only, or both postal code and street address.	√	√
P	Postal code match. Acquirer sent both postal code and street address, but street address not verified due to incompatible formats.	√	√
R	Retry: System unavailable or timed out. Issuer ordinarily performs its own AVS but was unavailable. Available for U.S. issuers only.	√	
S	Not applicable. If present, replaced with G (for international) or U (for domestic) by V.I.P. Available for U.S. Issuers only.	√	
U	Address not verified for domestic transaction. Visa tried to perform check on issuer's behalf but no AVS information was available on record, issuer is not an AVS participant, or AVS data was present in the request but issuer did not return an AVS result.	√	
W	Not applicable. If present, replaced with "Z" by V.I.P. Available for U.S. issuers only.	√	
X	Not applicable. If present, replaced with "Y" by V.I.P. Available for U.S. issuers only.	√	
Y	Street address and postal code match.	√	
Z	Postal/ZIP matches; street address does not match or street address not included in request.	√	√

Table of possible AVS response codes from Discover	
VALUE	DESCRIPTION
X	All digits match, nine-digit zip code.
A	All digits match, five-digit zip code.
Y	Address matches, zip code does not match.
T	Nine-digit zip code matches, address does not match.
Z	Five-digit zip codes matches, address does not match.
N	Nothing matches.
W	No data from issuer/authorization system.
U	Retry, system unable to process.
S	AVS not supported at this time.
Table of possible AVS response codes from American Express or JCB	
VALUE	DESCRIPTION
A	Billing address matches, postal code does not
D	Customer name incorrect, postal code matches
E	Customer name incorrect, billing address and postal code match
F	Customer name incorrect, billing address matches
K	Customer name matches
L	Customer name and postal code match
M	Customer name, billing address, and postal code match
N	Billing address and postal code do not match
O	Customer name and billing address match
R	System unavailable; retry
S	AVS not currently supported
U	Information is unavailable
W	Customer name, billing address, and postal code are all correct
Y	Billing address and postal code both match
Z	Postal Code matches, billing address does not

***For additional information on how to handle these responses, please refer to Appendix H. Additional Information for CVD and AVS.**

20. Appendix H. Additional Information for CVD and AVS

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

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



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

Also, please note that AVS verification is only applicable towards Visa, MasterCard, Discover and American Express transactions. This verification method is not applicable towards any other card type.

21. Appendix I. Sample Receipt

ON
www.eselectplus.ca

TRANSACTION APPROVED - THANK YOU

Payment Details

Transaction Type: PURCHASE
Transaction Amount: \$10.00 CAD
Order ID: Sample Receipt
Card Num: **** * 4242
Card Type: VISA
Resp Code - ISO Code: 027 - 01
Auth Code: 359047
Reference Num: 660064100011061260 M
Date/Time: Apr 06 2011 12:34PM

SIGNATURE
Cardholder will pay card issuer above amount pursuant to Cardholder Agreement

Item Details

Description	Product Code	Quantity	Price
Med Circle	Cir-001	1	\$2.00
Bir Triangle	Tri-002	1	\$1.00
Small Square	Squ-003	2	\$1.00
	Shipping:		\$1.00
	Tax 1:		\$1.00
	Tax 2:		\$1.00
	Total CAD:		\$10.00

Customer Details

Customer ID: 1234567890
Email Address:
Note:

Address Details

Billing	Shipping
Test Customer	Test Customer
Moneris Solutions	Moneris Solutions
123 Main Street	123 Main Street
Toronto	Toronto
On	On
M1M1M1	M1M1M1
Canada	Canada
Phone: 416-111-1111	Phone: 416-111-1111
Fax: 416-555-1111	Fax: 416-555-1111

eSELECTplus™

Copyright Notice

Copyright © 2012 Moneris Solutions, 3300 Bloor Street West, Toronto, Ontario, M8X 2X2

All Rights Reserved. This manual shall not wholly or in part, in any form or by any means, electronic, mechanical, including photocopying, be reproduced or transmitted without the authorized, written consent of Moneris Solutions.

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

Trademarks

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

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

Printed in Canada.

10 9 8 7 6 5 4 3 2 1