BE PAYMENT READY

Batch Upload Merchant Integration Guide

.xml

Canada only

Version 1.2.1 - August 2018

Table of Contents

Getting Help

Moneris has help for you at every stage of the integration process.

Getting Started	During Development	Production
Contact our Client Integration Specialists: clientintegrations@moneris.com Hours: Monday – Friday, 8:30am to 8 pm ET	If you are already working with an integration specialist and need technical development assistance, contact our eProducts Technical Consultants: 1-866-319-7450 eproducts@moneris.com Hours: 8am to 8pm ET	If your application is already live and you need production support, contact Moneris Customer Service: onlinepayments@moneris.com 1-866-319-7450 Available 24/7

For additional support resources, you can also make use of our community forums at

http://community.moneris.com/product-forums/

System and Skills Requirements

Before you start you will need to:

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

Additional requirements for Level 2/3 transactions:

- Batch files containing Level 2/3 transactions must use the .l23.xml extension

1 Introduction to Batch Upload

Moneris Gateway allows merchants the option of using a batch file to upload 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.

> **NOTE:** In no circumstances should CVD information be stored, and therefore should not be used with Batch Upload; if you need to process transactions using CVD information, use another solution such as the Moneris Gateway API or Moneris Hosted Solutions.

1.1 Implementing Your Batch Upload Solution

There are three main steps to batch file uploading via the Moneris Gateway:

1. **Creating a batch transaction file for uploading in the .xml format**. Transactions in batch files have specific structures that need to be conformed to. To learn more about this, see section 3 Sending Transactions and Receiving Responses.
2. **Configuring your SFTP client and connecting to Moneris Gateway**. Normally, configuration is only necessary to do once. To learn more about this,4.1 Configuring SFTP Client: What Do I Need to Do First?
3. **Uploading your batch transaction file and retrieving the response using your SFTP client**. To learn more about this, see 1 Uploading a Batch File and Receiving a Response.

These primary steps are nearly identical across the testing and production phases.

To learn more about testing, see 6 Testing Your Batch Upload Solution

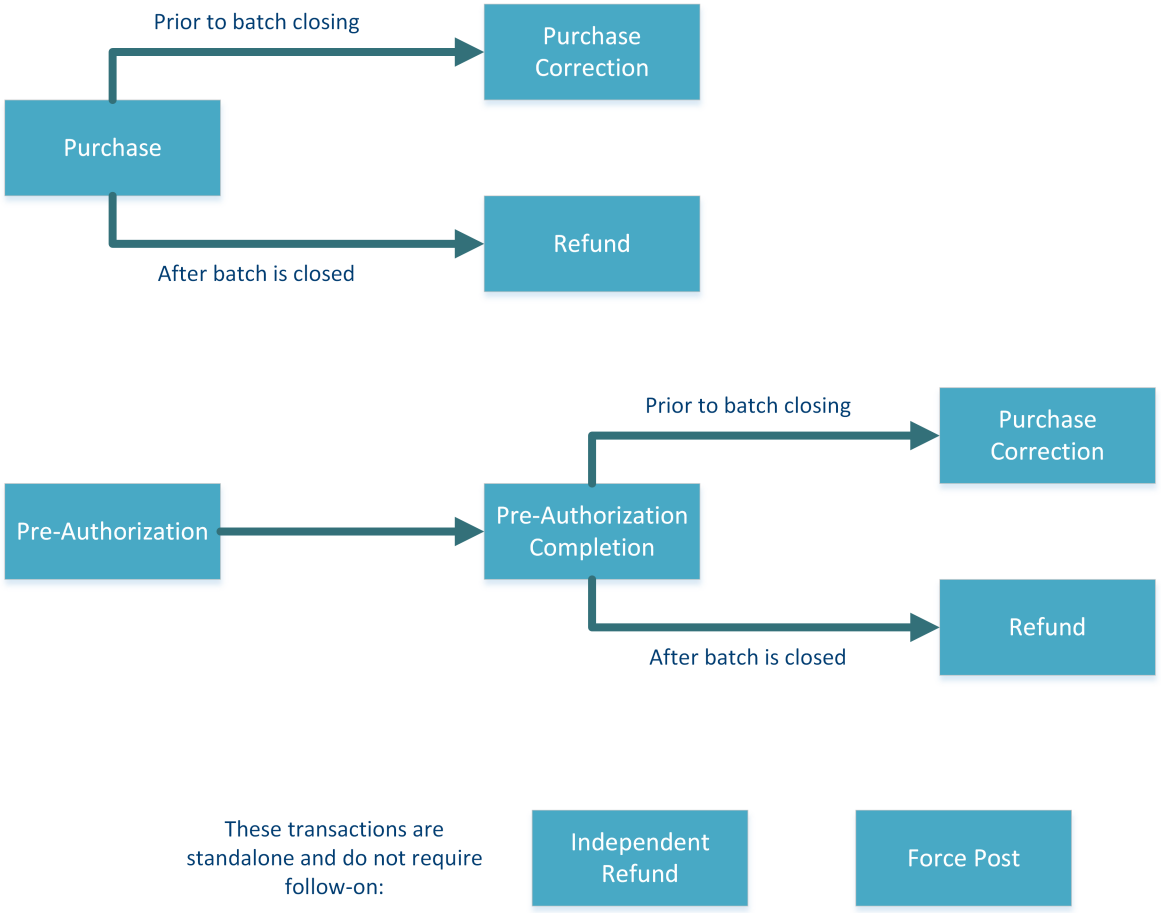
To learn more about production, see 7 Moving to Production

2 Transaction Types and Transaction Process Flows

Moneris Gateway supports a wide variety of transactions using the batch file upload method. Following are a list and brief descriptions of the transaction types supported.

2.1 Basic Transactions

2.1.1 Basic Transactions Process Flow



2.1.2 Purchase

A Purchase verifies funds on the customer's card, removes the funds and prepares them for deposit into the merchant's account.

Optional Features

Customer ID
Customer Information – See 2.7 Customer Information
Recurring Billing – See below

XML Parent Tag

```
<purchase>
```

Transaction Request Variables – Purchase

Table 1: Required Fields - Purchase transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99 **NOTE:** For Purchase or Purchase with Vault requests that include Recurring Billing, the value of this field is the amount to bill immediately.
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric	Expiry date with no spaces or slashes.

Variable and Field Name	Type and Limits	Description
	YYMM format	**NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-Commerce Indicator possible values: 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
Credential on File Info `cof_info`	*Object* n/a	Required for transactions using stored cardholder credentials This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.

Optional Transaction Request Variables – Purchase

Table 2: Optional Request Fields - Purchase transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes

Variable and Field Name	Type and Limits	Description
		EXAMPLE: policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Customer Information `cust_info`	*Object* n/a	This is a nested object within the transaction. See the Customer Information Request Fields table below for definitions of the Customer Information-specific fields.
Recurring Billing `recur`	*Object* n/a	Recurring Billing allows you to set up payments whereby Moneris automatically processes the transactions and bills customers on your behalf based on the billing cycle information you provide. This is a nested object within the transaction. See the Recurring Billing Transactions Request Fields table below for definitions of the Recurring Billing-specific fields.

Recurring Billing Transactions Request Fields

These are required only as part of Recurring Billing transaction requests.

XML Parent Tag

`<recur>`

Recurring Billing fields below are children of the `<recur>` tag

Table 3: Required Fields - Recurring Billing Transactions

Variable and Field Name	Type and Limits	Description
recur_unit	*String* day, week, month, eom	The unit that you wish to use as a basis for the Interval. This can be set as day, week or

Variable and Field Name	Type and Limits	Description
		month. Then using the "period" field you can configure how many days, weeks, months between billing cycles.
period	*String* 0 – 999 characters, numeric	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	*String* 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	*String* 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

Variable and Field Name	Type and Limits	Description
		billed on a regular basis thereafter.
recur_amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	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 999999.99. This is the amount that will be billed on the start_date and every interval thereafter.
num_recurs	*String* 1 – 99 characters, numeric	The number of times to recur the transaction.

Customer Information Request Fields

For information about the Customer Information object fields, see 2.7.2 Customer Information (cust_info) Fields.

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only sub-

Variable and Field Name	Type and Limits	Description
		sequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

2.1.3 Pre-Authorization

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 that have been locked by a Pre-Authorization transaction so that they may be settled in the merchant's account, a Pre-Authorization Completion transaction must be performed.

Optional Features

Customer Information – See 2.7 Customer Information

XML Parent Tag

`<preauth>`

Transaction Request Variables – Pre-Authorization

Table 1: Required Fields – Pre-Authorization transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.

Variable and Field Name	Type and Limits	Description
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce indicator possible values: 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
Credential on File Info `cof_info`	*Object*	Required for transactions using stored cardholder credentials This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.

Optional Transaction Request Variables – Pre-Authorization

Table 2: Optional Request Fields - Pre-Authorization transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number.

Variable and Field Name	Type and Limits	Description
		Can be searched from the Moneris Merchant Resource Center
Customer Information `cust_info`	*Object* n/a	This is a nested object within the transaction. See the Customer Information Request Fields table below for definitions of the Customer Information-specific fields.

Customer Information Request Fields

For information about the Customer Information object fields, see 2.7.2 Customer Information (cust_info) Fields.

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials

Variable and Field Name	Type and Limits	Description
		Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

2.1.4 Pre-Authorization Completion

Retrieves funds that have been locked by a Pre-Authorization transaction, and prepares them for settlement into the merchant's account.

XML Parent Tag

`<completion>`

Transaction Request Variables – Pre-Authorization Completion

Table 1: Required Fields – Pre-Authorization Completion transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount	*String*	Amount of the transaction.

Variable and Field Name	Type and Limits	Description
amount	9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Transaction number txn_number	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-Commerce Indicator possible values: 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

2.1.5 Force Post

Retrieves the locked funds and prepares them for settlement into the merchant's account.

This is used when a merchant obtains the authorization number directly from the issuer by a third-party authorization method (such as by phone).

> **NOTE:** This transaction is not supported for UnionPay

XML Parent Tag

`<forcepost>`

Table 1: Required Fields – Force Post transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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

Variable and Field Name	Type and Limits	Description
		been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
Authorization code `auth_code`	*String* 8-character alphanumeric	Authorization code provided in the transaction response from the issuing bank
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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

Optional Transaction Request Variables – Force Post

Table 2: Optional Request Fields – Force Post transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center

2.1.6 Purchase Correction

Restores the full amount of a previous Purchase, Force Post or Pre-Authorization Completion transaction to the cardholder's card, and removes any record of it from the cardholder's statement.

If the batch has already closed, use a Refund instead.

XML Parent Tag

`<purchasecorrection>`

Transaction Request Variables – Purchase Correction

Table 1: Required Fields – Purchase Correction transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Transaction number `txn_number`	*String* 255-character alphanumeric	Used when performing follow on transactions — this must be filled with the value that was returned as the txn_number in the response

Variable and Field Name	Type and Limits	Description
		of the original transaction. When performing a Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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

2.1.7 Refund

Restores all or part of the funds from a Purchase, Pre-Authorization Completion or Force Post transaction to the cardholder's card. If the transaction is still in an open batch please refer to Purchase Correction. Unlike a Purchase Correction, after a Refund there is a record of both the initial charge and the refund on the cardholder's statement.

XML Parent Tag

```
<refund>
```

Transaction Request Variables – Refund

Table 1: Required Fields – Refund transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Transaction number `txn_number`	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-Commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring

Variable and Field Name	Type and Limits	Description
		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

2.1.8 Independent Refund

Credits a specified amount to the cardholder's credit card.

It is not necessary for the transaction that you are refunding to have been processed via the Moneris Gateway.

XML Parent Tag

```
<ind_refund>
```

Transaction Request Variables – Independent Refund

Table 1: Required Fields – Independent Refund transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount	*String*	Amount of the transaction.

Variable and Field Name	Type and Limits	Description
`amount`	9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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

Variable and Field Name	Type and Limits	Description
		8 - Non-secure Transaction (Web or Email Based) 9 - SET non-authenticated transaction

Optional Transaction Request Variables – Independent Refund

Table 2: Optional Request Fields – Independent Refund transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center

2.2 Mag Swipe Transactions

Mag Swipe Purchase

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 Pre-Authorization

The Mag Swipe Pre-Authorization 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 Pre-Authorization so that they may be settled in the merchant's account a Mag Swipe Completion must be performed.

Mag Swipe Completion

Once a Mag Swipe Pre-Authorization is obtained the funds that are locked need to be retrieved from the customer's credit card. The Mag Swipe Completion retrieves the locked funds and readies them for settlement into the merchant's account.

Mag Swipe Purchase Correction

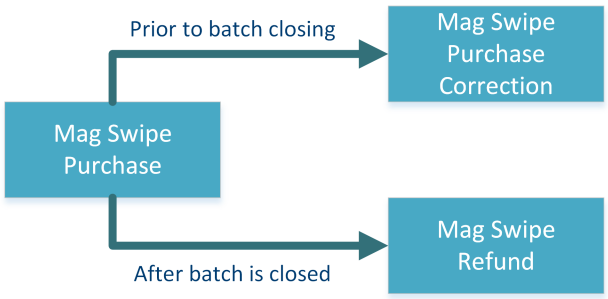
Mag Swipe Purchase and Mag Swipe Completion transactions can be voided the same day that they occur. A Mag Swipe Purchase Correction must be for the full amount of the transaction and will remove any record of it from the cardholder's statement.

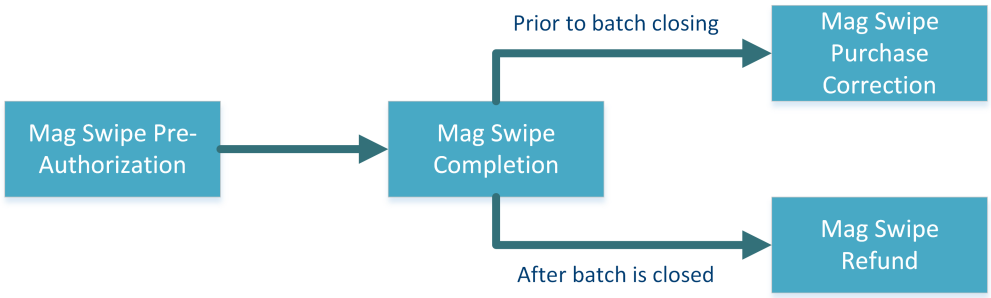
> **NOTE:** A Purchase Correction can be performed against a transaction as long as the batch that contains the original transaction remains open.

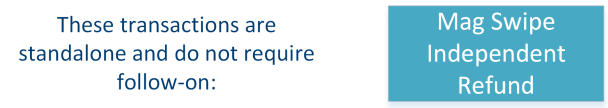
Mag Swipe Refund
can be performed against a Mag Swipe Purchase or a Mag Swipe Completion to refund any part, or all of the transaction.

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 Completion.

2.2.1 Mag Swipe Transactions Process Flow







2.2.2 Sending 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.

Table 1: Required Fields – Mag Swipe Transactions

Transaction Type	Fields
Mag Swipe Purchase (track2_purchase)	track2_purchase, order_id, cust_id, amount, track2, pan, expdate, pos_code
Mag Swipe Pre-Authorization (track2_preauth)	track2_preauth, order_id, cust_id, pan, expdate, pos_code
Mag Swipe Capture (track2_completion)	track2_completion, order_id(from track2_ preauth), comp_amount, txn_number (from track2_preauth)
Mag Swipe Void (track2_purchasecorrection)	track2_purchasecorrection, order_id(from original transaction), txn_number (from track2_purchase or track2_completion)
Mag Swipe Refund (track2_refund)	track2_refund,order_id (from original transaction), amount, txn_number (from track2_purchase or track2_completion)
Mag Swipe Independent Refund (track2_ind_refund)	track2_ind_refund, order_id, cust_id, amount, track2, pan, expdate, pos_code

2.2.2.1 Example – Batch Including Mag Swipe Transactions

The following sample code illustrates a batch file containing Mag Swipe and other transaction requests. A corresponding example for the transaction response follows.

> **NOTE:** In a Mag Swipe/track2 transaction, you must still include all of the required tags such as the 'pan' or 'expdate' fields.

```
<?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>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<purchase>
<order_id>test_xml_00002</order_id>
<amount>1.04</amount>
<pan>42424242424242</pan>
<expdate>0806</expdate>
<crypt_type>1</crypt_type>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<track2_purchase>
<order_id>track2_testing_purch_1</order_id>
<cust_id>my cust id</cust_id>
<amount>1.00</amount>
<track2>;5258984987184986=060610160910010606O2?</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=060610160910010606O2?</track2>
<pan></pan>
<expdate></expdate>
<pos_code>00</pos_code>
</track2_preauth>
</request>
```

Corresponding Example Response for Mag Swipe Transaction

```
<?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>
<IssuerId>123456789012345</IssuerId>
</receipt>
```

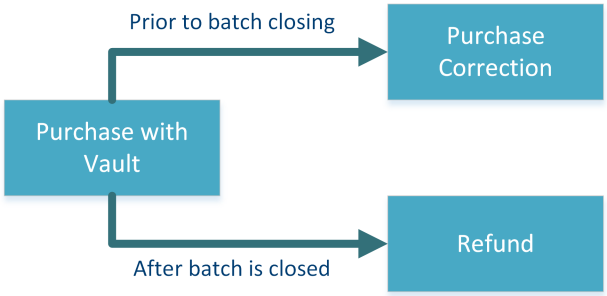
```
<receipt>
<ReceiptId>test_xml_00002</ReceiptId>
<ReferenceNum>660021630014070190</ReferenceNum>
<ResponseCode>075</ResponseCode>
<ISO>14</ISO>
<AuthCode>000000</AuthCode>
<TransTime>16:04:35</TransTime>
<TransDate>2006-06-20</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<Message>DECLINED * 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>
<IssuerId>123456789012345</IssuerId>
</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>
```

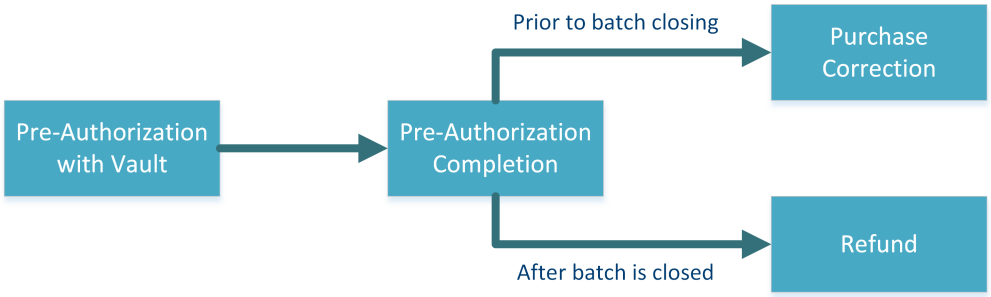
2.3 Vault Transactions

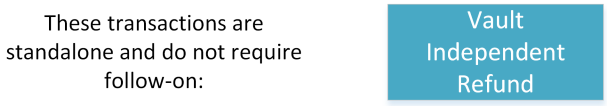
The Vault feature allows merchants to create customer profiles, edit those profiles, and use them to process transactions without having to enter financial information each time. Customer profiles store customer data essential to processing transactions, including credit and signature debit details.

The Vault is a complement to the recurring payment module. It securely stores customer account information on Moneris Solutions' secure servers. This allows merchants to bill customers for routine products or services when an invoice is due.

2.3.1 Vault Transactions Process Flow







2.3.2 Vault Administrative Transactions

2.3.2.1 About Vault Administrative Transactions

Vault Administrative transactions allow the user to perform such tasks as creating new Vault profiles and deleting existing profiles.

2.3.2.2 Vault Add Credit Card

Creates a new credit card profile, and generates a unique data key which can be obtained from the Receipt object. This data key is the profile identifier that all future financial Vault transactions will use to associate with the saved information.

XML Parent Tag

`<res_add_cc>`

Transaction Request Variables – Vault Add Credit Card

Table 1: Required Fields – Vault Add Credit Card transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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

Variable and Field Name	Type and Limits	Description
		been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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
Credential on File Info `cof_info`	*Object*	Required for transactions using stored cardholder credentials This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.

Optional Transaction Request Variables – Vault Add Credit Card

Table 2: Optional Fields – Vault Add Credit Card transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Phone Number `phone`	*String* 30-character alphanumeric	Phone number of the customer
Note `note`	*String* 30-character alphanumeric	Used for supplementary information

Variable and Field Name	Type and Limits	Description
Email address `email`	*String* 30-character alphanumeric	Email address of the customer

2.3.2.3 Vault Update Credit Card

Updates a Vault profile (based on the data key) to contain credit card information. All information contained within a credit card profile is updated as indicated by the submitted fields.

> **NOTE:** The Credential on File information object is required for this transaction *only* when updating the credit card number; for all other situations, do not include the Credential on File info object or its indicator fields in the request.

XML Parent Tag

`<res_update_cc>`

Transaction Request Variables – Vault Update Credit Card

Table 1: Required Fields – Vault Update Credit Card transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Data key `data_key`	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card trans-

Variable and Field Name	Type and Limits	Description
		action) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered

Optional Transaction Request Variables – Vault Update Credit Card

Table 2: Optional Fields – Vault Update Credit Card transaction

Variable and Field Name	Type and Limits	Description
Credential on File Info `cof_info` **NOTE:** The Credential on File Info object is required when updating a credit card number; in all other cases, do not send it	*Object* n/a	Required for transactions using stored cardholder credentials Must be included if the credit card number (`pan`) is sent This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.

Variable and Field Name	Type and Limits	Description
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Phone Number `phone`	*String* 30-character alphanumeric	Phone number of the customer
Note `note`	*String* 30-character alphanumeric	Used for supplementary information
Email address `email`	*String* 30-character alphanumeric	Email address of the customer

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.

2.3.2.4 Vault Delete

Deletes an existing Vault profile of any type using the unique data key that was assigned when the profile was added

NOTE: After a profile is deleted, the information that was contained in it can no longer be retrieved.

XML Parent Tag

`<res_delete>`

Transaction Request Variables – Vault Delete

Table 1: Required Fields – Vault Update Credit Card transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Author-

Variable and Field Name	Type and Limits	Description
		ization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Data key `data_key`	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered

2.3.2.5 Example – Batch With Vault Admin Transactions

```
<?xml version="1.0"?>
<request>
<store_id>store3</store_id>
<api_token>yesguy</api_token>
<res_add_cc>
<order_id>Moneris_test_Dec20_1</order_id>
<pan>4242424242424242</pan>
<expdate>1212</expdate>
<crypt_type>7</crypt_type>
<cof_info>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</res_add_cc>
<res_add_cc>
<order_id>Moneris_test_Dec20_2</order_id>
<pan>5454545454545454</pan>
<expdate>1201</expdate>
<crypt_type>7</crypt_type>
<cof_info>
<issuer_id>123456789012345</issuer_id>
</cof_info>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
```

```
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
</res_add_cc>
<res_update_cc>
<order_id>Moneris_test_Dec20_3</order_id>
<data_key>XKwBrc6n8sHIJftK1mNf0TFb3</data_key>
<expdate>1212</expdate>
</res_update_cc>
<res_update_cc>
<order_id>Moneris_test_Dec20_4</order_id>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<pan>5454545454545454</pan>
<expdate>1201</expdate>
<crypt_type>7</crypt_type>
<cof_info>
<issuer_id>123456789012345</issuer_id>
</cof_info>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
</res_update_cc>
<res_delete>
<order_id>Moneris_test_Dec20_5</order_id>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
</res_delete>
</request>
```

Corresponding Example Response -for Vault Admin Transaction

```
<?xml version="1.0"?>
<response>
<receipt>
<DataKey>wjxVIIFcJAqKhBZ3URASTO8MO</DataKey>
<ReceiptId>Moneris_test_Dec20_1</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>001</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<Message>Successfully registered CC details.</Message>
<TransTime>14:02:19</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id></cust_id>
<phone></phone>
<email></email>
<note></note>
<crypt_type>7</crypt_type>
<masked_pan>4242***4242</masked_pan>
<expdate>1212</expdate>
</ResolveData>
<issuer_id>123456789012345</issuer_id>
```

```
</receipt>
<receipt>
<DataKey>NrFoDGGF6QqtDGVKiVm6etwp3</DataKey>
<ReceiptId>Moneris_test_Dec20_2</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>001</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<Message>Successfully registered CC details.</Message>
<TransTime>14:02:29</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<crypt_type>7</crypt_type>
<masked_pan>5454***5454</masked_pan>
<expdate>1201</expdate>
</ResolveData>
<issuer_id>123456789012345</issuer_id>
</receipt>
<receipt>
<DataKey>XKwBrc6n8sHIJftK1mNf0TFb3</DataKey>
<ReceiptId>Moneris_test_Dec20_3</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>001</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<Message>Successfully updated CC details.</Message>
<TransTime>14:02:39</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id></cust_id>
<phone></phone>
<email></email>
<note></note>
<crypt_type>7</crypt_type>
<expdate>1212</expdate>
<masked_pan>4242***4242</masked_pan>
</ResolveData>
</receipt>
```

```
<receipt>
<DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
<ReceiptId>Moneris_test_Dec20_4</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>001</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<Message>Successfully updated CC details.</Message>
<TransTime>14:02:49</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<avs_street_name>main st</avs_street_name>
<avs_street_number>1</avs_street_number>
<avs_zipcode>123456</avs_zipcode>
<crypt_type>7</crypt_type>
<expdate>1201</expdate>
<masked_pan>5454***5454</masked_pan>
</ResolveData>
<issuer_id>123456789012345</issuer_id>
</receipt>
<receipt>
<DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
<ReceiptId>Moneris_test_Dec20_5</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>001</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<Message>Successfully deleted CC details.</Message>
<TransTime>16:44:20</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>null</CorporateCard>
<RecurSuccess>null</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<expdate>1201</expdate>
<avs_street_number>1</avs_street_number>
<avs_street_name>main st</avs_street_name>
```

```
<avs_zipcode>123456</avs_zipcode>
<masked_pan>5454***5454</masked_pan>
<crypt_type>7</crypt_type>
</ResolveData>
</receipt>
</response>
```

2.3.3 Vault Financial Transactions

- 2.3.3.2 Purchase with Vault
- 1 Purchase with Vault and Credential on File
- 2.3.3.3 Pre-Authorization with Vault
- 1 Pre-Authorization with Vault and Credential on File
- 2.3.3.4 Vault Independent Refund

2.3.3.1 About Vault Financial Transactions

The Vault feature allows merchants to create long-term customer profiles, edit those profiles, and use them to process transactions without having to enter financial information each time.

2.3.3.2 Purchase with Vault

This transaction has been deprecated for future use; going forward, use Purchase with Vault and Credential on File.

This transaction uses the data key to identify a previously registered credit card profile. The details saved within the profile are then submitted to perform a Purchase transaction.

Optional Features

Recurring Billing – See below
Customer Information – See 2.7 Customer Information

XML Parent Tag

`<res_purchase_cc>`

Transaction Request Variables – Purchase with Vault

Table 1: Required Fields – Purchase with Vault transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique

Variable and Field Name	Type and Limits	Description
		for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Data key `data_key`	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment

Variable and Field Name	Type and Limits	Description
		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
Credential on File Info `cof_info`	*Object* n/a	Required for transactions using stored cardholder credentials This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.

Table 2: Optional Fields – Purchase with Vault transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Customer Information `cust_info`	*Object* n/a	This is a nested object within the transaction. See the Customer Information Request Fields table below for definitions of the Customer Information-specific fields.
Recurring Billing	*Object*	Recurring Billing allows you to set up payments whereby Mon-

Variable and Field Name	Type and Limits	Description
recur	n/a	eris automatically processes the transactions and bills customers on your behalf based on the billing cycle information you provide. This is a nested object within the transaction. See the Recurring Billing Transasctions Request Fields table below for definitions of the Recurring Billing-specific fields.

Recurring Billing Transactions Request Fields

These are required only as part of Recurring Billing transaction requests.

XML Parent Tag

<recur>

Recurring Billing fields below are children of the <recur> tag

Table 3: Required Fields - Recurring Billing Transactions

Variable and Field Name	Type and Limits	Description
recur_unit	*String* day, week, month, eom	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	*String* 0 – 999 characters, numeric	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.

Variable and Field Name	Type and Limits	Description
		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	*String* 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	*String* 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	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	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 999999.99. This is the amount that will be billed on the start_date and every interval thereafter.
num_recurs	*String* 1 – 99 characters, numeric	The number of times to recur the transaction.

Customer Information Request Fields

For information about the Customer Information object fields, see 2.7.2 Customer Information (cust_info) Fields.

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are:

Variable and Field Name	Type and Limits	Description
		2 - subsequent transactions (using previously stored payment details)

2.3.3.3 Pre-Authorization with Vault

This transaction has been deprecated for future use; going forward, use Pre-Authorization with Vault and Credential on File.

This transaction uses the data key to identify a previously registered credit card profile. The details saved within the profile are then submitted to perform a Pre-Authorization transaction.

Optional Features

Customer Information – See 2.7 Customer Information

XML Parent Tag

`<res_preauth_cc>`

Transaction Request Variables – Pre-Authorization with Vault

Table 1: Required Fields – Pre-Authorization with Vault transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can

Variable and Field Name	Type and Limits	Description
	EXAMPLE: 123456.78	be 0.01 and the maximum 999999.99
Data key `data_key`	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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
Credential on File Info `cof_info`	*Object* n/a	Required for transactions using stored cardholder credentials This is a nested object within

Variable and Field Name	Type and Limits	Description
		the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.

Table 2: Optional Fields – Pre-Authorization with Vault transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Customer Information `cust_info`	*Object* n/a	This is a nested object within the transaction. See the Customer Information Request Fields table below for definitions of the Customer Information-specific fields.

Customer Information Request Fields

For information about the Customer Information object fields, see 2.7.2 Customer Information (cust_info) Fields.

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id`	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials

Variable and Field Name	Type and Limits	Description
NOTE: This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).		Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

2.3.3.4 Vault Independent Refund

Credits a specified amount to the cardholder's credit card. The credit card number and expiry date are mandatory.

It is not necessary for the transaction that you are refunding to have been processed via the Moneris Gateway.

XML Parent Tag

```
<res_ind_refund_cc>
```

Transaction Request Variables – Vault Independent Refund

Table 1: Required Fields – Vault Independent Refund transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Data key `data_key`	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order -

Variable and Field Name	Type and Limits	Description
		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

Table 2: Optional Fields – Vault Independent Refund transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center

2.3.3.5 Example – Batch With Vault Financial Transactions

The following sample code illustrates a batch file that includes both Vault and basic financial transactions , followed by a sample of the corresponding response.

```
<?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>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<purchase>
<order_id>test_xml_00002</order_id>
<amount>1.04</amount>
<pan>42424242424242</pan>
<expdate>0806</expdate>
<crypt_type>1</crypt_type>
</purchase>
<res_purchase_cc>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<order_id>Moneris_test_Dec20_6</order_id>
<cust_id>My_Customer_Name</cust_id>
<amount>55.00</amount>
<crypt_type>7</crypt_type>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</res_purchase_cc>
</request>
```

Corresponding Example Response for Vault Financial Transaction

```
<?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>
<IssuerId>123456789012345</IssuerId>
</receipt>
<receipt>
<ReceiptId>test_xml_00002</ReceiptId>
<ReferenceNum>660021630014070190</ReferenceNum>
<ResponseCode>075</ResponseCode>
<ISO>14</ISO>
<AuthCode>000000</AuthCode>
<TransTime>16:04:35</TransTime>
<TransDate>2006-06-20</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<Message>DECLINED * CARD PROBLEM =CARD PROBLEM</Message>
<TransAmount>1.04</TransAmount>
```

```
<CardType>00</CardType>
<TransID>156091-19-0</TransID>
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
</receipt>
<receipt>
<DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
<ReceiptId>Moneris_test_Dec20_6</ReceiptId>
<ReferenceNum>660021820010918800</ReferenceNum>
<ResponseCode>027</ResponseCode>
<ISO>01</ISO>
<AuthCode>001945</AuthCode>
<Message>APPROVED * =</Message>
<TransTime>14:03:53</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<TransAmount>55.00</TransAmount>
<CardType>M</CardType>
<TransID>966830-0_7</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>false</CorporateCard>
<RecurSuccess>false</RecurSuccess>
<AvsResultCode>null</AvsResultCode>
<CvdResultCode>null</CvdResultCode>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<expdate>1201</expdate>
<avs_street_number>1</avs_street_number>
<avs_street_name>main st</avs_street_name>
<avs_zipcode>123456</avs_zipcode>
<masked_pan>5454***5454</masked_pan>
<crypt_type>7</crypt_type>
</ResolveData>
<IssuerId>123456789012345</IssuerId>
</receipt>
</response>
```

2.4 Level 2/3 Transactions

Level 2/3 Purchase – (batch_l23purchase)

The Batch Upload Level 2/3 Purchase handles Pre-Authorization and Pre-Authorization Completion in a single step. If you have already done a Pre-Authorization, perform a Level 2/3 Force Post instead.

Level 2/3 Force Post – (batch_l23forcepost)

The Level 2/3 Force Post retrieves the locked funds from a previously processed Pre-Authorization and readies them for settlement into the merchant's account along with the submission of level 2/3 details sent within the transaction request.

Level 2/3 Purchase Correction – (batch_l23purchasecorrection)

Purchase and Force Post transactions 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. This transaction type does not accept Level 2/3 details as no information is submitted to bank.

NOTE: A Purchase Correction can be performed against a transaction as long as the batch that contains the original transaction remains open.

Level 2/3 Refund – (batch_l23refund)

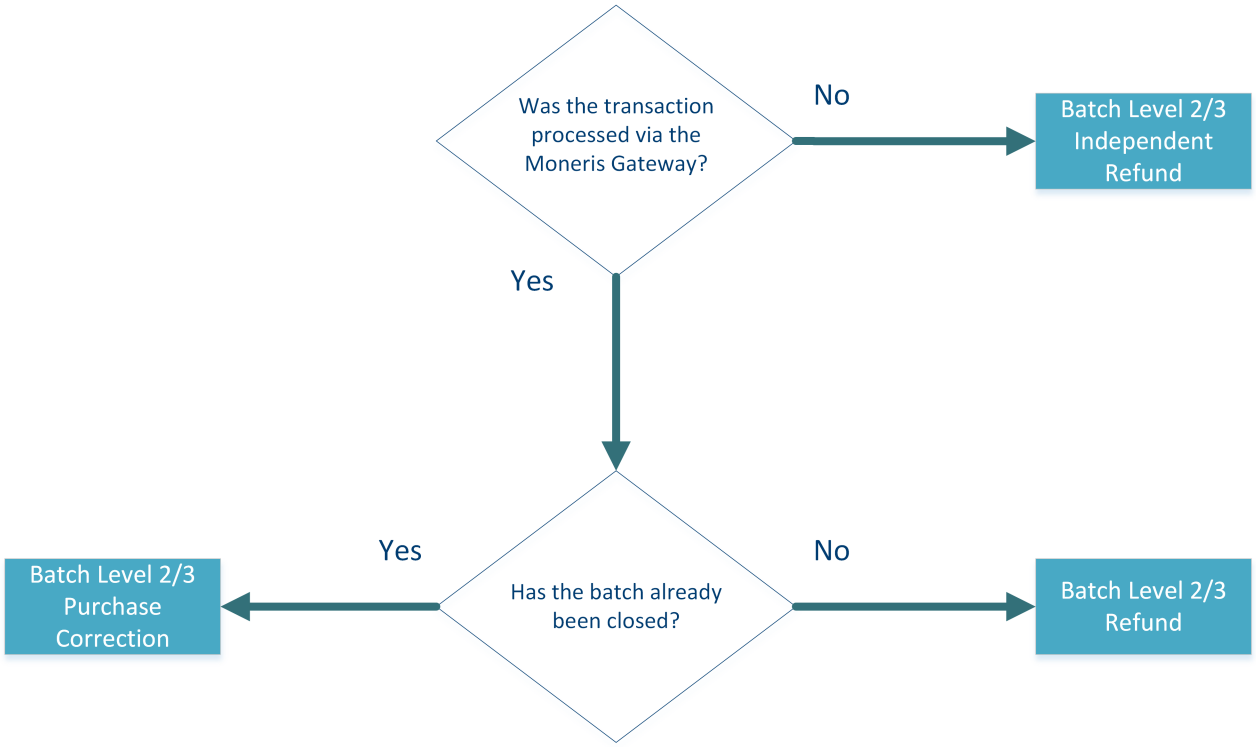
A Refund with level 2/3 details can be performed against a Purchase or a Force Post to refund any part, or all of the transaction. This refund will show up on cardholder's statement along with level 2/3 details that were submitted with this transaction request. A prior Purchase or Force Post transaction is required to use this transaction.

Level 2/3 Independent Refund – (batch_l23ind_refund)

An Independent Refund can be performed to credit money to a Credit Card along with the submission of level 2/3 details pertaining to this refund. This transaction does not require a prior Purchase or Force Post.

2.4.1 Transaction Flow for Level 2/3 Batch Upload

2.4.1.1 Batch Upload Level 2/3 Refunds Process Flow



2.4.2 Sending Level 2/3 Transactions

For Level 2/3 transactions, the basic Pre-Authorization transaction is used to authorize payment from a customer, and if the response contains `CorporateCard = true`, then the Level 2/3 details will be captured in the following Level 2/3 Force Post transaction.

Level 2/3 Force Post and Level 2/3 Independent Refund transactions do not require prior transaction requests.

Batch files that contain Level 2/3 transactions must be sent in a different batch file than non-Level 2/3 transactions, using the filename extension .l23.xml.

The following tables illustrates the fields that you can send for each transaction type. For a full description of each field, please refer to Appendix A Definitions of Required Fields.

For an example including all of the transaction types and their respective structures, please refer to 2.4.8 Example – XML for Level 2/3 Transaction.

XML Root Tag

```
<l23_batch>
```

> **NOTE:** The root tag for Level 2/3 transactions is the <l23_batch>, and the <request> tag is the child of <l23_batch>.

> **NOTE:** The above mentioned transactions can only be performed on purchasing corporate card. Addendum1 is mandatory. Also at least 1 or more Addendum2 is required. If account is setup with Level 2/3 Addendum Data bypass account rights, merchant may send in Non Purchasing Corporate Cards with addendum 1 and 2. In this scenario, the transaction will be processed as a Level 1 transaction but the entire request XML including addendum 1 and 2 will be echoed back in the transaction response.

2.4.3 Level 2/3 Purchase – Batch Upload

XML Parent Tag – Level 2/3 Purchase

```
<batchl23_purchase>
```

Store ID

```
<store_id>

    <subchild>
```

Table 1: Required Fields – Batch Upload Level 2/3 Purchase transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator	*String*	E-commerce Indicator possible values:

Variable and Field Name	Type and Limits	Description
`crypt_type`	1-character alphanumeric	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
Credential on File Info `cof_info`	*Object*	Required whenever storing cardholder credentials for the first time and when using those credentials in subsequent transactions. This is a nested object within the transaction. See the Credential on File Request Fields table below for definitions of the Credential on File-specific fields.
Addendum 1 (Level 2) `addendum1`	*Object* N/A	Contains Level 2 field details; see Addendum 1 Request Fields table below for fields

Table 2: Optional Request Fields – Level 2/3 Purchase transaction

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes

Variable and Field Name	Type and Limits	Description
		EXAMPLE: policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Addendum 2 (Level 3) `addendum2`	*Object* N/A	Contains Level 3 field details; see Addendum 2 Request Fields table below for fields

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated

Variable and Field Name	Type and Limits	Description
		transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

Request Fields for Level 2/3 Addendum Objects

Table 3: Addendum 1 Request Fields – Level 2 Details

Variable and Field Name	Type and Limits	Description
Customer code `customer_code`	*String* 40-character alphanumeric	A control number, such as purchase order number, project number, department allocation number or name supplied by the customer
Discount amount `discount_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Monetary amount of discount on the transaction
Freight amount `freight_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Shipping cost on the transaction

Variable and Field Name	Type and Limits	Description
Duty amount `duty_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Applicable amount of duty charged on the transaction
National tax amount `national_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of federal-level taxes charged on the transaction, e.g., GST or HST
Other tax amount `other_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Provincial tax amount charged on the transaction, e.g., PST or QST
Merchant VAT number `merchant_vat_no`	*String* 15-character alphanumeric	Your GST registration number
Customer VAT number `customer_vat_no`	*String* 13-character alphanumeric	The customer's GST registration number
Destination postal code `destination_zip`	*String* 10-character alphanumeric	Postal code of the destination of goods
Ship from postal code `ship_from_zip`	*String* 10-character alphanumeric	Postal code of the shipping location

Table 4: Addendum 2 Request Fields – Level 3 Details

Variable and Field Name	Type and Limits	Description
Item description `item_description`	*String* 40-character alphanumeric	Description of item; length allowed may differ by card brand
Product code `product_code`	*String* 12-character alphanumeric	UPC or product code of an item
Quantity `quantity`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Quantity of units of a particular item being purchased
Unit cost `unit_cost`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Unit cost of an item
Total cost `ext_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Total cost of a particular item being purchased (quantity X unit cost)
Unit of measure `uom`	*String* 12-character alphanumeric	Unit of measure used for the item being purchased, as in common use for international

Variable and Field Name	Type and Limits	Description
		trade
Item discount amount `item_discount_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of discount applied to total cost of item
Item tax amount `item_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of national-level tax charged on item; for Canada, this is GST or HST.
Item other tax amount `item_other_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of any other taxes applied to the item purchased
Item other tax type `item_other_tax_type`	*String* 4-character alphanumeric	Indicates the type of other tax being applied Possible values: PST - Provincial Sales Tax QST - Quebec Sales Tax (or other)

2.4.4 Level 2/3 Force Post – Batch Upload

XML Parent Tag – Level 2/3 Force Post

```
<batchl23_forcepost>
```

Store ID
```
<store_id>

    <subchild>
```

Table 1: Required Fields – Batch Upload Level 2/3 Force Post transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
Transaction number `txn_number`	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the

Variable and Field Name	Type and Limits	Description
		Purchase.
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
Authorization code `auth_code`	*String* 8-character alphanumeric	Authorization code provided in the transaction response from the issuing bank
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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

Variable and Field Name	Type and Limits	Description
Addendum 1 (Level 2) `addendum1`	*Object* N/A	Contains Level 2 field details; see Addendum 1 Request Fields table below for fields

Optional Transaction Request Variables – Level 2/3 Force Post

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Addendum 2 (Level 3) `addendum2`	*Object* N/A	Contains Level 3 field details; see Addendum 2 Request Fields table below for fields

Request Fields for Level 2/3 Addendum Objects

Table 2: Addendum 1 Request Fields – Level 2 Details

Variable and Field Name	Type and Limits	Description
Customer code `customer_code`	*String* 40-character alphanumeric	A control number, such as purchase order number, project number, department allocation number or name supplied by the customer
Discount amount `discount_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point	Monetary amount of discount on the transaction

Variable and Field Name	Type and Limits	Description
	EXAMPLE: 123456.78	
Freight amount `freight_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Shipping cost on the transaction
Duty amount `duty_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Applicable amount of duty charged on the transaction
National tax amount `national_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of federal-level taxes charged on the transaction, e.g., GST or HST
Other tax amount `other_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Provincial tax amount charged on the transaction, e.g., PST or QST

Variable and Field Name	Type and Limits	Description
Merchant VAT number `merchant_vat_no`	*String* 15-character alphanumeric	Your GST registration number
Customer VAT number `customer_vat_no`	*String* 13-character alphanumeric	The customer's GST registration number
Destination postal code `destination_zip`	*String* 10-character alphanumeric	Postal code of the destination of goods
Ship from postal code `ship_from_zip`	*String* 10-character alphanumeric	Postal code of the shipping location

Table 3: Addendum 2 Request Fields – Level 3 Details

Variable and Field Name	Type and Limits	Description
Item description `item_description`	*String* 40-character alphanumeric	Description of item; length allowed may differ by card brand
Product code `product_code`	*String* 12-character alphanumeric	UPC or product code of an item
Quantity `quantity`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Quantity of units of a particular item being purchased
Unit cost `unit_cost`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point	Unit cost of an item

Variable and Field Name	Type and Limits	Description
	EXAMPLE: 123456.78	
Total cost `ext_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Total cost of a particular item being purchased (quantity X unit cost)
Unit of measure `uom`	*String* 12-character alphanumeric	Unit of measure used for the item being purchased, as in common use for international trade
Item discount amount `item_discount_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of discount applied to total cost of item
Item tax amount `item_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of national-level tax charged on item; for Canada, this is GST or HST.
Item other tax amount `item_other_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the	Dollar amount of any other taxes applied to the item purchased

Variable and Field Name	Type and Limits	Description
	decimal point **EXAMPLE:** 123456.78	
Item other tax type `item_other_tax_type`	*String* 4-character alphanumeric	Indicates the type of other tax being applied Possible values: PST - Provincial Sales Tax QST - Quebec Sales Tax (or other)

2.4.5 Level 2/3 Purchase Correction – Batch Upload

Restores the full amount of a previous Level 2/3 Purchase or Level 2/3 Force Post transaction to the cardholder's card, and removes any record of it from the cardholder's statement.

If the batch has already closed, use a Level 2/3 Refund instead.

XML Parent Tag – Level 2/3 Purchase Correction

```
<batchl23_purchasecorrection>
```

Store ID
`<store_id>`

`<subchild>`

Transaction Request Variables – Level 2/3 Purchase Correction

Table 1: Required Fields – Level 2/3 Purchase Correction transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as

Variable and Field Name	Type and Limits	Description
		that of the original transaction.
Transaction number `txn_number`	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-commerce Indicator possible values: 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

2.4.6 Level 2/3 Refund – Batch Upload

Restores all or part of the funds from a Level 2/3 Purchase or Level 2/3 Force Post transaction to the cardholder's card. Unlike a Purchase Correction, after a Refund there is a record of both the initial charge and the refund on the cardholder's statement.

If the transaction is still in an open batch, use the Level 2/3 Purchase Correction instead.

XML Parent Tag – Level 2/3 Refund

```
<batchl23_refund>
```

Store ID
```
<store_id>

    <subchild>
```

Transaction Request Variables – Level 2/3 Refund

Table 1: Required Fields – Level 2/3 Refund transaction

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Transaction number `txn_number`	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.

Variable and Field Name	Type and Limits	Description
E-commerce indicator `crypt_type`	*String* 1-character alphanumeric	E-Commerce Indicator possible values: 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
Addendum 1 (Level 2) `addendum1`	*Object* N/A	Contains Level 2 field details; see Addendum 1 Request Fields table below for fields

Optional Request Fields – Level 2/3 Refund

Table 2: Optional Request Fields – Level 2/3 Refund transaction

Variable and Field Name	Type and Limits	Description
Addendum 2 (Level 3) `addendum2`	*Object* N/A	Contains Level 3 field details; see Addendum 2 Request Fields table below for fields

Request Fields for Level 2/3 Addendum Objects

Table 3: Addendum 1 Request Fields – Level 2 Details

Variable and Field Name	Type and Limits	Description
Customer code `customer_code`	*String* 40-character alphanumeric	A control number, such as purchase order number, project number, department alloc-

Variable and Field Name	Type and Limits	Description
		ation number or name supplied by the customer
Discount amount `discount_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Monetary amount of discount on the transaction
Freight amount `freight_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Shipping cost on the transaction
Duty amount `duty_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Applicable amount of duty charged on the transaction
National tax amount `national_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of federal-level taxes charged on the transaction, e.g., GST or HST

Variable and Field Name	Type and Limits	Description
Other tax amount `other_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Provincial tax amount charged on the transaction, e.g., PST or QST
Merchant VAT number `merchant_vat_no`	*String* 15-character alphanumeric	Your GST registration number
Customer VAT number `customer_vat_no`	*String* 13-character alphanumeric	The customer's GST registration number
Destination postal code `destination_zip`	*String* 10-character alphanumeric	Postal code of the destination of goods
Ship from postal code `ship_from_zip`	*String* 10-character alphanumeric	Postal code of the shipping location

Table 4: Addendum 2 Request Fields – Level 3 Details

Variable and Field Name	Type and Limits	Description
Item description `item_description`	*String* 40-character alphanumeric	Description of item; length allowed may differ by card brand
Product code `product_code`	*String* 12-character alphanumeric	UPC or product code of an item
Quantity `quantity`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point	Quantity of units of a particular item being purchased

Variable and Field Name	Type and Limits	Description
	EXAMPLE: 123456.78	
Unit cost `unit_cost`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Unit cost of an item
Total cost `ext_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Total cost of a particular item being purchased (quantity X unit cost)
Unit of measure `uom`	*String* 12-character alphanumeric	Unit of measure used for the item being purchased, as in common use for international trade
Item discount amount `item_discount_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of discount applied to total cost of item
Item tax amount `item_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the	Dollar amount of national-level tax charged on item; for Canada, this is GST or HST.

Variable and Field Name	Type and Limits	Description
	decimal point **EXAMPLE:** 123456.78	
Item other tax amount `item_other_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of any other taxes applied to the item purchased
Item other tax type `item_other_tax_type`	*String* 4-character alphanumeric	Indicates the type of other tax being applied Possible values: PST - Provincial Sales Tax QST - Quebec Sales Tax (or other)

2.4.7 Level 2/3 Independent Refund – Batch Upload

Credits a specified amount to the cardholder's credit card, while also including Level 2 and/or Level 3 information.

It is not necessary for the transaction that you are refunding to have been processed via the Moneris Gateway.

XML Parent Tag – Level 2/3 Independent Refund

```
<batchl23_ind_refund>
```

Store ID

```
<store_id>

    <subchild>
```

Transaction Request Variables – Level 2/3 Independent Refund

Variable and Field Name	Type and Limits	Description
Order ID `order_id`	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount `amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99
Credit card number `pan`	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Expiry Date `expdate`	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator	*String*	E-commerce Indicator possible values:

Variable and Field Name	Type and Limits	Description
`crypt_type`	1-character alphanumeric	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
Addendum 1 (Level 2) `addendum1`	*Object* N/A	Contains Level 2 field details; see Addendum 1 Request Fields table below for fields

Optional Transaction Request Variables – Level 2/3 Independent Refund

Variable and Field Name	Type and Limits	Description
Customer ID `cust_id`	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Addendum 2 (Level 3) `addendum2`	*Object* N/A	Contains Level 3 field details; see Addendum 2 Request Fields table below for fields

Request Fields for Level 2/3 Addendum Objects

Table 1: Addendum 1 Request Fields – Level 2 Details

Variable and Field Name	Type and Limits	Description
Customer code `customer_code`	*String* 40-character alphanumeric	A control number, such as purchase order number, project number, department allocation number or name supplied by the customer
Discount amount `discount_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Monetary amount of discount on the transaction
Freight amount `freight_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Shipping cost on the transaction
Duty amount `duty_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Applicable amount of duty charged on the transaction
National tax amount `national_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal	Amount of federal-level taxes charged on the transaction, e.g., GST or HST

Variable and Field Name	Type and Limits	Description
	point **EXAMPLE:** 123456.78	
Other tax amount `other_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Provincial tax amount charged on the transaction, e.g., PST or QST
Merchant VAT number `merchant_vat_no`	*String* 15-character alphanumeric	Your GST registration number
Customer VAT number `customer_vat_no`	*String* 13-character alphanumeric	The customer's GST registration number
Destination postal code `destination_zip`	*String* 10-character alphanumeric	Postal code of the destination of goods
Ship from postal code `ship_from_zip`	*String* 10-character alphanumeric	Postal code of the shipping location

Table 2: Addendum 2 Request Fields – Level 3 Details

Variable and Field Name	Type and Limits	Description
Item description `item_description`	*String* 40-character alphanumeric	Description of item; length allowed may differ by card brand
Product code `product_code`	*String* 12-character alphanumeric	UPC or product code of an item
Quantity	*String*	Quantity of units of a particular item being purchased

Variable and Field Name	Type and Limits	Description
`quantity`	9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	
Unit cost `unit_cost`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Unit cost of an item
Total cost `ext_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Total cost of a particular item being purchased (quantity X unit cost)
Unit of measure `uom`	*String* 12-character alphanumeric	Unit of measure used for the item being purchased, as in common use for international trade
Item discount amount `item_discount_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of discount applied to total cost of item

Variable and Field Name	Type and Limits	Description
Item tax amount `item_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of national-level tax charged on item; for Canada, this is GST or HST.
Item other tax amount `item_other_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of any other taxes applied to the item purchased
Item other tax type `item_other_tax_type`	*String* 4-character alphanumeric	Indicates the type of other tax being applied Possible values: PST - Provincial Sales Tax QST - Quebec Sales Tax (or other)

2.4.8 Example – XML for Level 2/3 Transaction

The following sample code illustrates an XML request for a Level 2/3 transaction. The example response subsequently follows.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<l23_batch>
    <request>
        <store_id>moneris</store_id>
        <api_token>hurgle</api_token>
        <batch_l23purchase>
            <order_id>NQA_20140428_0</order_id>
            <cust_id>custid123</cust_id>
            <amount>1.00</amount>
            <pan>4242424254545454</pan>
            <expdate>1512</expdate>
            <crypt_type>7</crypt_type>
            <addendum1>
                <customer_code>customer_code</customer_code>
```

```
            <discount_amount>1.12</discount_amount>
            <freight_amount>2.23</freight_amount>
            <duty_amount>3.34</duty_amount>
            <national_tax_amount>4.56</national_tax_amount>
            <other_tax_amount>5.67</other_tax_amount>
            <merchant_vat_no>merchant_vat_no</merchant_vat_no>
            <customer_vat_no>cus_vat_no</customer_vat_no>
            <destination_zip>M1R 1W4</destination_zip>
            <ship_from_zip>M6K 2H9</ship_from_zip>
          </addendum1>
        <addendum2>
            <item_description>item_description</item_description>
            <product_code>product_code</product_code>
            <quantity>12</quantity>
            <unit_cost>1.2345</unit_cost>
            <ext_amount>2.89</ext_amount>
            <uom>EA</uom>
            <item_discount_amount>6.78</item_discount_amount>
            <item_tax_amount>4.32</item_tax_amount>
            <item_other_tax_amount>1.94</item_other_tax_amount>
            <item_other_tax_type>QST</item_other_tax_type>
        </addendum2>
    </batch_l23purchase>
</request>
<request>
    <store_id>moneris</store_id>
    <api_token>hurgle</api_token>
    <batch_l23ind_refund>
        <order_id>NQA_20140428_1</order_id>
        <cust_id>custid123</cust_id>
        <amount>1.00</amount>
        <pan>4242424254545454</pan>
        <expdate>1512</expdate>
        <crypt_type>7</crypt_type>
        <addendum1>
            <customer_code>customer_code</customer_code>
            <discount_amount>1.12</discount_amount>
            <freight_amount>2.23</freight_amount>
            <duty_amount>3.34</duty_amount>
            <national_tax_amount>4.56</national_tax_amount>
            <other_tax_amount>5.67</other_tax_amount>
            <merchant_vat_no>merchant_vat_no</merchant_vat_no>
            <customer_vat_no>cus_vat_no</customer_vat_no>
            <destination_zip>M1R 1W4</destination_zip>
            <ship_from_zip>M6K 2H9</ship_from_zip>
        </addendum1>
        <addendum2>
            <item_description>item_description</item_description>
            <product_code>product_code</product_code>
            <quantity>12</quantity>
            <unit_cost>1.2345</unit_cost>
            <ext_amount>2.89</ext_amount>
            <uom>EA</uom>
            <item_discount_amount>6.78</item_discount_amount>
            <item_tax_amount>4.32</item_tax_amount>
            <item_other_tax_amount>1.94</item_other_tax_amount>
            <item_other_tax_type>QST</item_other_tax_type>
        </addendum2>
    </batch_l23ind_refund>
    <batch_l23forcepost>
        <order_id>NQA_20140428_3</order_id>
        <cust_id>custid123</cust_id>
        <amount>1.00</amount>
        <pan>4242424254545454</pan>
        <expdate>1512</expdate>
        <auth_code>964734</auth_code>
```

```
            <crypt_type>7</crypt_type>
            <addendum1>
                <customer_code>customer_code</customer_code>
                <discount_amount>1.12</discount_amount>
                <freight_amount>2.23</freight_amount>
                <duty_amount>3.34</duty_amount>
                <national_tax_amount>4.56</national_tax_amount>
                <other_tax_amount>5.67</other_tax_amount>
                <merchant_vat_no>merchant_vat_no</merchant_vat_no>
                <customer_vat_no>cus_vat_no</customer_vat_no>
                <destination_zip>M1R 1W4</destination_zip>
                <ship_from_zip>M6K 2H9</ship_from_zip>
            </addendum1>
            <addendum2>
                <item_description>item_description</item_description>
                <product_code>product_code</product_code>
                <quantity>12</quantity>
                <unit_cost>1.2345</unit_cost>
                <ext_amount>2.89</ext_amount>
                <uom>EA</uom>
                <item_discount_amount>6.78</item_discount_amount>
                <item_tax_amount>4.32</item_tax_amount>
                <item_other_tax_amount>1.94</item_other_tax_amount>
                <item_other_tax_type>QST</item_other_tax_type>
            </addendum2>
        </batch_l23forcepost>
    </request>
</l23_batch>
```

Corresponding Response Example

NOTE: The response below has been re-formatted with spacing and newlines for documentation purpose. In the actual response, the whole response is a single line, so you should be parsing for XML tags rather than newline.

```
<l23_batch>
    <response>
        <orig_request>
            <request>
                <store_id>moneris</store_id>
                <api_token>hurgle</api_token>
                <batch_l23purchase>
                    <order_id>NQA_20140428_0</order_id>
                    <cust_id>custid123</cust_id>
                    <amount>1.00</amount>
                    <pan>4242-****5454</pan>
                    <expdate>1512</expdate>
                    <crypt_type>7</crypt_type>
                    <addendum1>
                        <customer_code>customer_code</customer_code>
                        <discount_amount>1.12</discount_amount>
                        <freight_amount>2.23</freight_amount>
                        <duty_amount>3.34</duty_amount>
                        <national_tax_amount>4.56</national_tax_amount>
                        <other_tax_amount>5.67</other_tax_amount>
                        <merchant_vat_no>merchant_vat_no</merchant_vat_no>
                        <customer_vat_no>cus_vat_no</customer_vat_no>
                        <destination_zip>M1R 1W4</destination_zip>
```

```
                    <ship_from_zip>M6K 2H9</ship_from_zip>
                </addendum1>
                <addendum2>
                    <item_description>item_description</item_description>
                    <product_code>product_code</product_code>
                    <quantity>12</quantity>
                    <unit_cost>1.2345</unit_cost>
                    <ext_amount>2.89</ext_amount>
                    <uom>EA</uom>
                    <item_discount_amount>6.78</item_discount_amount>
                    <item_tax_amount>4.32</item_tax_amount>
                    <item_other_tax_amount>1.94</item_other_tax_amount>
                    <item_other_tax_type>QST</item_other_tax_type>
                </addendum2>
            </batch_l23purchase>
        </request>
    </orig_request>
    <receipt>
        <ReceiptId>NQA_20140428_0</ReceiptId>
        <ReferenceNum>660117310013420270</ReferenceNum>
        <ResponseCode>005</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>996941</AuthCode>
        <TransTime>16:14:44</TransTime>
        <TransDate>2014-04-28</TransDate>
        <TransType>02</TransType>
        <Complete>true</Complete>
        <Message>APPROVED           *                    =: Level 2/3 message have been stored
in database.</Message>
        <TransAmount>1.00</TransAmount>
        <CardType>V</CardType>
        <TransID>97634-1_9</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
        <CorporateCard>true</CorporateCard>
        <MessageId>1A4118584849702</MessageId>
    </receipt>
</response>
<response>
    <orig_request>
        <request>
            <store_id>moneris</store_id>
            <api_token>hurgle</api_token>
            <batch_l23ind_refund>
                <order_id>NQA_20140428_1</order_id>
                <cust_id>custid123</cust_id>
                <amount>1.00</amount>
                <pan>4242-****5454</pan>
                <expdate>1512</expdate>
                <crypt_type>7</crypt_type>
                <addendum1>
                    <customer_code>customer_code</customer_code>
                    <discount_amount>1.12</discount_amount>
                    <freight_amount>2.23</freight_amount>
                    <duty_amount>3.34</duty_amount>
                    <national_tax_amount>4.56</national_tax_amount>
                    <other_tax_amount>5.67</other_tax_amount>
                    <merchant_vat_no>merchant_vat_no</merchant_vat_no>
                    <customer_vat_no>cus_vat_no</customer_vat_no>
                    <destination_zip>M1R 1W4</destination_zip>
                    <ship_from_zip>M6K 2H9</ship_from_zip>
                </addendum1>
                <addendum2>
                    <item_description>item_description</item_description>
                    <product_code>product_code</product_code>
```

```
                    <quantity>12</quantity><unit_cost>1.2345</unit_cost>
                    <ext_amount>2.89</ext_amount>
                    <uom>EA</uom>
                    <item_discount_amount>6.78</item_discount_amount>
                    <item_tax_amount>4.32</item_tax_amount>
                    <item_other_tax_amount>1.94</item_other_tax_amount>
                    <item_other_tax_type>QST</item_other_tax_type>
                </addendum2>
            </batch_l23ind_refund>
        </request>
    </orig_request>
    <receipt>
        <ReceiptId>NQA_20140428_1</ReceiptId>
        <ReferenceNum>660117310013420280</ReferenceNum>
        <ResponseCode>005</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>010237</AuthCode>
        <TransTime>16:14:45</TransTime>
        <TransDate>2014-04-28</TransDate>
        <TransType>04</TransType>
        <Complete>true</Complete>
        <Message>APPROVED           *                    =: Level 2/3 message have been stored
in database.</Message>
        <TransAmount>1.00</TransAmount>
        <CardType>V</CardType>
        <TransID>97635-0_9</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
        <CorporateCard>true</CorporateCard>
        <MessageId>1A4118584859712</MessageId>
    </receipt>
</response>
<response>
    <orig_request>
        <request>
            <store_id>moneris</store_id>
            <api_token>hurgle</api_token>
            <batch_l23forcepost>
                <order_id>NQA_20140428_3</order_id>
                <cust_id>custid123</cust_id>
                <amount>1.00</amount>
                <pan>4242-****5454</pan>
                <expdate>1512</expdate>
                <auth_code>964734</auth_code>
                <crypt_type>7</crypt_type>
                <addendum1>
                    <customer_code>customer_code</customer_code>
                    <discount_amount>1.12</discount_amount>
                    <freight_amount>2.23</freight_amount>
                    <duty_amount>3.34</duty_amount>
                    <national_tax_amount>4.56</national_tax_amount>
                    <other_tax_amount>5.67</other_tax_amount>
                    <merchant_vat_no>merchant_vat_no</merchant_vat_no>
                    <customer_vat_no>cus_vat_no</customer_vat_no>
                    <destination_zip>M1R 1W4</destination_zip>
                    <ship_from_zip>M6K 2H9</ship_from_zip>
                </addendum1>
                <addendum2>
                    <item_description>item_description</item_description>
                    <product_code>product_code</product_code>
                    <quantity>12</quantity>
                    <unit_cost>1.2345</unit_cost>
                    <ext_amount>2.89</ext_amount>
                    <uom>EA</uom>
                    <item_discount_amount>6.78</item_discount_amount>
```

```
                        <item_tax_amount>4.32</item_tax_amount>
                        <item_other_tax_amount>1.94</item_other_tax_amount>
                        <item_other_tax_type>QST</item_other_tax_type>
                    </addendum2>
                </batch_l23forcepost>
            </request>
        </orig_request>
        <receipt>
            <ReceiptId>NQA_20140428_3</ReceiptId>
            <ReferenceNum>660117310013420290</ReferenceNum>
            <ResponseCode>005</ResponseCode>
            <ISO>01</ISO>
            <AuthCode>964734</AuthCode>
            <TransTime>16:14:46</TransTime>
            <TransDate>2014-04-28</TransDate>
            <TransType>02</TransType>
            <Complete>true</Complete>
            <Message>APPROVED           *                    =: Level 2/3 message have been stored
    in database.</Message>
            <TransAmount>1.00</TransAmount>
            <CardType>V</CardType>
            <TransID>97636-0_9</TransID>
            <TimedOut>false</TimedOut>
            <BankTotals>null</BankTotals>
            <Ticket>null</Ticket>
            <CorporateCard>true</CorporateCard>
            <MessageId>1A4118584869720</MessageId>
        </receipt>
    </response>
</l23_batch>
```

2.5 Credential on File

2.5.1 About Credential on File Transactions – Batch Upload

The Credential on File object is used to send cardholder credentials in the following circumstances:

- Whenever a cardholder's credentials are being stored for the first time
- On all subsequent transactions where stored credentials are being used to process a transaction

All Batch Upload transactions involving Purchase, Pre-Authorization and Recurring Billing are considered Credential on File transactions.

Transactions where cardholder credentials are being stored for the first time are not able to be processed using the Batch Upload method.

Before processing Batch Upload Credential on File transactions, you must use another processing method to store the credentials, including the Moneris Gateway API, Moneris Hosted Solutions, or via the Moneris Merchant Resource Center. When using these other methods, you can either perform a financial transaction, or else use the Card Verification transaction to store the cardholder's credentials without charging them.

For more information on processing transactions with the Moneris Gateway API, Hosted Solutions or Merchant Resource Center, please refer to the Moneris Developer Portal at https://developer-.moneris.com.

2.5.2 Batch Upload Transactions with Credential on File

The following Batch Upload transaction types are impacted by Credential on File rules:

Purchase with Credential on File
Purchase with Vault and Credential on File
Purchase with Recurring Billing and Credential on File
Pre-Authorization with Credential on File
Pre-Authorization with Vault and Credential on File
Vault Add Credit Card with Credential on File
Vault Update Credit Card with Credential on File

2.5.3 Credential on File Info Object Request Fields

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials

Variable and Field Name	Type and Limits	Description
		Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

2.6 Recurring Billing

2.6.1 About Recurring Billing Transactions – Batch Upload

Recurring Billing transactions are essentially purchase transactions that repeat multiple times, e.g., memberships. When set up as recurring, an additional set of "recur" variables are added to the following transaction types:

Purchase (purchase)
Vault Purchase (res_purchase_cc)

2.6.2 Sending Recurring Transactions

When uploading a recurring transaction you will need to indicate:

- the recurring amount (as represented by recur_amount,
- interval (period),

- start date (start_date)
- the number of times it will recur (num_recurs).

There is also an option to bill a different amount immediately.

> **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.

2.6.3 Example – Batch With Recurring Transactions

The following sample code illustrates a batch file that includes Purchase transactions with Recurring Billing information, followed by a sample of the corresponding response.

```
<?xml version="1.0"?>
<request>
<store_id>store1</store_id>
<api_token>yesguy</api_token>
<purchase>
<order_id>464571415531_1</order_id>
<amount>3.00</amount>
<pan>4242424242424242</pan>
<expdate>0803</expdate>
<crypt_type>7</crypt_type>
<recur>
<recur_unit>week</recur_unit>
<start_now>true</start_now>
<start_date>2011/10/30</start_date>
<num_recurs>4</num_recurs>
<period>2</period>
<recur_amount>1.00</recur_amount>
</recur>
<cof_info>
<payment_indicator>R </payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<purchase>
<order_id>464571415532_2</order_id>
<amount>1.00</amount>
<pan>4242424242424242</pan>
<expdate>0903</expdate>
<crypt_type>7</crypt_type>
<recur>
<recur_unit>day</recur_unit>
<start_now>false</start_now>
<start_date>2011/09/15</start_date>
<num_recurs>10</num_recurs>
<period>30</period>
<recur_amount>1.00</recur_amount>
</recur>
<cof_info>
<payment_indicator>R</payment_indicator>
```

```
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<purchase>
<order_id>464571415533_3</order_id>
<amount>5.00</amount>
<pan>4242424242424242</pan>
<expdate>0703</expdate>
<crypt_type>7</crypt_type>
<recur>
<recur_unit>month</recur_unit>
<start_now>true</start_now>
<start_date>2011/11/11</start_date>
<num_recurs>12</num_recurs>
<period>1</period>
<recur_amount>1.00</recur_amount>
</recur>
<cof_info>
<payment_indicator>R</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>
<res_purchase_cc>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<order_id>Moneris_test_Dec20_4</order_id>
<cust_id>My_Customer_Name</cust_id>
<amount>55.00</amount>
<crypt_type>7</crypt_type>
<recur>
<recur_unit>month</recur_unit>
<start_now>true</start_now>
<start_date>2011/12/01</start_date>
<num_recurs>12</num_recurs>
<period>1</period>
<recur_amount>1.00</recur_amount>
</recur>
<cof_info>
<payment_indicator>R</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</res_purchase_cc>
</request>
```

Corresponding Example Response - Recurring Transactions

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

```
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
<RecurSuccess>true</RecurSuccess>
<IssuerId>123456789012345</IssuerId>
</receipt>
<receipt>
<ReceiptId>464571415532_2</ReceiptId>
<ReferenceNum>null</ReferenceNum>
<ResponseCode>null</ResponseCode>
<ISO>null</ISO>
<AuthCode>null</AuthCode>
<TransTime>null</TransTime>
<TransDate>null</TransDate>
<TransType>null</TransType>
<Complete>true</Complete>
<Message>Recurring transaction successfully registered.</Message>
<TransAmount>null</TransAmount>
<CardType>null</CardType>
<TransID>null</TransID>
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
<RecurSuccess>true</RecurSuccess>
<IssuerId>123456789012345</IssuerId>
</receipt>
<receipt>
<ReceiptId>464571415533_3</ReceiptId>
<ReferenceNum>660021730013780250</ReferenceNum>
<ResponseCode>027</ResponseCode>
<ISO>01</ISO>
<AuthCode>010212</AuthCode>
<TransTime>16:24:41</TransTime>
<TransDate>2006-06-20</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<Message>APPROVED * =: Recurring transaction successfully registered.</Message>
<TransAmount>5.00</TransAmount>
<CardType>V</CardType>
<TransID>84119-25-0</TransID>
<TimedOut>false</TimedOut>
<BankTotals>null</BankTotals>
<Ticket>null</Ticket>
<RecurSuccess>true</RecurSuccess>
<IssuerId>123456789012345</IssuerId>
</receipt>
<receipt>
<DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
<ReceiptId>Moneris_test_Dec20_4</ReceiptId>
<ReferenceNum>660021820010918800</ReferenceNum>
<ResponseCode>027</ResponseCode>
<ISO>01</ISO>
<AuthCode>001945</AuthCode>
<Message>APPROVED * =: Recurring transaction successfully registered.</Message>
<TransTime>14:03:53</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<TransAmount>55.00</TransAmount>
<CardType>M</CardType>
<TransID>966830-0_7</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>false</CorporateCard>
<RecurSuccess>true</RecurSuccess>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
```

```
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<expdate>1201</expdate>
<masked_pan>5454***5454</masked_pan>
<crypt_type>7</crypt_type>
</ResolveData>
<IssuerId>123456789012345</IssuerId>
</receipt>
</response>
```

2.7 Customer Information

2.7.1 About Customer Info Transactions – Batch Upload

You can also upload transactions with additional customer information, represented under the <cust_info> tag,with a Purchase or Pre-Authorization transactions via the XML batch upload.

Additional customer information includes:

- Billing and shipping information - details about the customer such as names, addresses, etc.
- Item information - details about the product being purchased
- Other details, such as email or special instructions

When uploading customer information, all of the fields within cust_info must be included.

2.7.2 Customer Information (cust_info) Fields

Billing and Shipping Information

The following table lists the fields related to billing and shipping information. The fields for billing and shipping information, contained in the <billing> and <shipping> tags, are identical and therefore are repeated in both tags.

Table 1: Billing and Shipping Information – Transaction with Customer and Order Details

Field Name	Size/Type
first_name	30-character alphanumeric
last_name	30-character alphanumeric
company_name	30-character alphanumeric
address	30-character alphanumeric
city	30-character alphanumeric
province	30-character alphanumeric
postal_code	30-character alphanumeric
country	30-character alphanumeric
phone_number	30-character alphanumeric
fax	30-character alphanumeric
tax1	30-character alphanumeric
tax2	30-character alphanumeric
tax3	30-character alphanumeric
shipping_cost	30-character alphanumeric

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.

Table 2: Item Information – Transaction with Customer and Order Details

Field Name	Size/Type
name	30-character alphanumeric
quantity	10-character numeric
product_code	30-character alphanumeric
extended_amount	9-character decimal

Other Details

Table 3: Other Details – Transaction with Customer and Order Details

Field Name	Size/Type
email	50-character alphanumeric
instructions	50-character alphanumeric

NOTE:

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

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

2.7.3 Example Transaction with Customer Information

The following sample code illustrates a Purchase transaction with all optional fields: cust_id as well as cust_info. All fields within cust_info have been populated and two items have been included.

```
<?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>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>

</request>
```

Corresponding Response Example - Transaction with Customer Information

```
<?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>
<issuer_id>123456789012345</issuer_id>
</receipt>
</response>
```

2.7.4 Example Transaction with Customer Info and Vault

The following sample code illustrates a batch Purchase request with two transaction types (one basic transaction and one using Vault) with customer information, followed by the corresponding response 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>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</purchase>

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

```
<product_code>shoes_101</product_code>
<extended_amount>1.00</extended_amount>
</item>
<item>
<name>Blue Suede Shoes</name>
<quantity>10</quantity>
<product_code>shoes_102</product_code>
<extended_amount>3.00</extended_amount>
</item>
</cust_info>
<cof_info>
<payment_indicator>U</payment_indicator>
<payment_information>2</payment_information>
<issuer_id>123456789012345</issuer_id>
</cof_info>
</res_purchase_cc>

</request>
```

Corresponding Example Response for Transaction with Customer Information and Vault

```
<?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>
<issuer_id>123456789012345</issuer_id>
</receipt>
<receipt>
<DataKey>RbFbmySFugYNRSmTaN6Gkm08r</DataKey>
<ReceiptId>Moneris_test_Dec20_6</ReceiptId>
<ReferenceNum>660021820010918800</ReferenceNum>
<ResponseCode>027</ResponseCode>
<ISO>01</ISO>
<AuthCode>001945</AuthCode>
<Message>APPROVED * =</Message>
<TransTime>14:03:53</TransTime>
<TransDate>2010-12-21</TransDate>
<TransType>00</TransType>
<Complete>true</Complete>
<TransAmount>55.00</TransAmount>
<CardType>M</CardType>
<TransID>966830-0_7</TransID>
<TimedOut>false</TimedOut>
<CorporateCard>false</CorporateCard>
<RecurSuccess>false</RecurSuccess>
<ResSuccess>true</ResSuccess>
<PaymentType>cc</PaymentType>
<ResolveData>
<cust_id>My_Customer_Name</cust_id>
```

```
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
<expdate>1201</expdate>
<avs_street_number>1</avs_street_number>
<avs_street_name>main st</avs_street_name>
<avs_zipcode>123456</avs_zipcode>
<masked_pan>5454***5454</masked_pan>
<crypt_type>7</crypt_type>
</ResolveData>
<issuer_id>123456789012345</issuer_id>
</receipt>
</response>
```

3 Sending Transactions and Receiving Responses

- 3.1 Creating a Batch File
- 1 What Information Do I Need to Include in a Transaction Request?

A batch upload transaction actually consists of two parts:

1. Sending the transaction request to the Moneris Gateway, contained in a batch .xml file, and
2. Receiving the transaction response from the Moneris Gateway, contained in a response file.

You send transaction files and receive their responses using SFTP, and so before you start sending transactions or receiving their responses you need to configure your SFTP client.

To learn more about configuring your SFTP client, see "Configuring SFTP Client: What Do I Need to Do First?" on page 109.

3.1 Creating a Batch File

When you are creating a file for batch upload, it must follow these standards:

- The file name must be alphanumeric
- It cannot contain any spaces
- The extension must be .xml, and
- It must be lowercase

File names that do not meet these requirements will not be processed.

Batch files can contain multiple transaction types. For information about the transaction types in Batch Upload, see 2 Transaction Types and Transaction Process Flows.

Batch files that contain Level 2/3 transactions must be sent in a different batch file than non-Level 2/3 transactions, using the filename extension .l23.xml.

3.2 XML Structure of Batch Transaction Requests

Batch Upload transaction requests are structured with a `<request>` root tag, followed by its immediate children:

`<store_id>`

this is your Moneris Store ID

`<api_token>`

this is the API token generated when you create your Store ID

`<[transaction type]>`

this is the transaction request you are sending, e.g., Purchase, Pre-Authorization, etc. Each transaction type has its own sub-children (the transaction variables).

The Batch Upload XML request structure looks like this:

```
<request>
        <store_id>...</store_id>
        <api token>...</api_token>
        <[transaction type]> ... </[transaction type]>
                <[variable1]> ... <[variable1]>


                 ...


        </request>
```

This structure is slightly modified when sending Level 2/3 requests. For information about XML structure of Level 2/3 transactions, see 3.2.1 XML Structure of Level 2/3 Transaction Requests.

3.2.1 XML Structure of Level 2/3 Transaction Requests

Level 2/3 transactions have a similar XML structure as other transactions, however, the <request> is encapsulated by a <l23_batch> root tag, as shown below. Level 2/3 transactions include two child tags, <addendum1> and <addendum2>, which contain variables for Level 2 and Level 3, respectively.

```
<l23_batch>
        <request>
                <store_id>...</store_id>
                <api token>...</api_token>
                <[transaction type]> ... </[transaction type]>
                        <variable>...</variable>
                        <addendum1>
                                <variable>...</variable>
                                ...
                        </addendum1>
                        <addendum2>
                                <variable>...</variable>
                                ...
                        <addendum2>
```

```
        </request>
</l23_batch>
```

4 Connecting to the Moneris Gateway to Upload Files

Once you have created a batch file, you need to connect to the Moneris Gateway to upload the file. This is done using a file transfer client that supports Secure File Transfer Protocol (SFTP). Transactions in both the testing and production environments are sent using a SFTP client of your choice.

4.1 Configuring SFTP Client: What Do I Need to Do First?

To send transactions, use the following credentials in your SFTP client to connect to the Moneris Gateway and log in.

	In the Test Environment	In the Production Environment
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 get your test and/or production SFTP username and password, please contact Moneris customer service at:

email: onlinepayments@moneris.com or call 1-866-319-7450

You will be asked for your merchant name, merchant number and store ID.

Once you are connected and logged in to the Moneris Gateway, you are ready to upload files for either testing or production purposes.

To learn more on this subject, see also:

5 Uploading a Batch File and Getting the Response

7 Moving to Production

5 Uploading a Batch File and Getting the Response

- 1 What Is Contained in a Transaction Response File?

When uploading a file it must be put in your default root directory. Do not put it in the processing directory, as 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 be appended with the extension ".out", as shown in the following table.

Table 1: Example of Request and Response Filename Usage

Request File	Response File
testing_june_20.xml	testing_june_20.xml.out

5.1 Determining Whether a Transaction Was Successful

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://developer-.moneris.com/More/Testing/Financial%20Response%20Codes.

6 Testing Your Batch Upload Solution

A testing environment is available for you to connect to while you are testing your batch upload solution. The test environment is generally available 24/7, however we cannot guarantee 100% availability.

> **NOTE:** Please be aware that other merchants are using the test environment so you may see batch files that you did not create. As a courtesy to others that are testing, we ask that when you are processing refunds, deleting files and/or other functions that you use only the transactions/files that you created.

To upload .xml batch files to the test environment, first you must enter the correct credentials in your SFTP client as described in 4.1 Configuring SFTP Client: What Do I Need to Do First?.

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 Center to view reports of your transactions.

To access the Merchant Resource Center in the test environment go to https://esqa.moneris.com/mpg and use the login credentials provided in the table below.

> **EXAMPLE:** If you have uploaded the batch file to store3, please log into the Merchant Resource Center into store3 to locate your transactions.

> **NOTE:** These test store IDs, API tokens, usernames and passwords are different than the credentials you use in the production environment.

Table 1: 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 or track2 data for Mag Swipe test transactions. Since this is a testing environment, please do not use any production cards or swipe any production cards for track2 data.

Table 2: Test Card Numbers

Card Plan	Test Card Number
MasterCard	5454545454545454
Visa	4242424242424242
Amex	373599005095005
Diners	36462462742008
Track2 (Mag Swipe)	;5258984987184986=06061016091001060602?

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.

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://developer-.moneris.com/More/Testing/Penny%20Value%20Simulator.

6.1 Test Store Credentials

For testing purposes, you can either use the pre-existing test stores, or you can create your own unique test store where you will only see your own transactions. If you are using the Moneris Merchant Resource Center 3.0 to manage your transaction activity, you *must* use your own unique test store and test API token. If you are still using an older version of the Merchant Resource Center and want to use pre-existing stores, use the test credentials provided in the following tables.

NOTE: For testing Batch Upload Level 2/3 transactions, use store2 only.

Table 1: Test Server Credentials - Canada

Store ID	API Token	MRC Username	MRC Password
store1	yesguy	demouser	password
store2	yesguy	demouser	password
store3	yesguy	demouser	password
store4	yesguy	demouser	password
store5	yesguy	demouser	password

Alternatively, you can create and use a unique test store where you will only see your own transactions. For more on this, see 1 Getting a Unique Test Store ID and API Token.

6.2 Test Card Numbers

Because of security and compliance reasons, the use of live credit and debit card numbers for testing is strictly prohibited. Only test credit and debit card numbers are to be used.

To test general transactions, use the following test card numbers:

Table 2: General test card numbers

Card Plan	Card Number
Mastercard	5454545454545454
Visa	4242424242424242
Amex	373599005095005
JCB	3566007770015365
Diners	36462462742008
Track2	5258968987035454=06061015454001060101?
Discover	6510000000000182
UnionPay	6250944000000771

6.3 Test Card Numbers for Level 2/3

When testing Level 2/3 transactions, use the card numbers below.

Card Brand	Test Card Number
Mastercard	5454545442424242
Visa	4242424254545454
Amex	373269005095005

7 Moving to Production

- 1 How Do I Activate My Store?

Once you have completed the necessary development and testing, you are ready to move your solution into production.

The process of sending transaction requests and receiving the responses is nearly identical in production as it is in testing. To review the primary steps in the batch upload process again, see "Implementing Your Batch Upload Solution" on page 6.

To start processing live transactions, you need to activate your production store.

> **NOTE:**
> To receive your Production SFTP Username and Password, please e-mail the Integration Support department at:
>
> onlinepayments@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.

7.1 Activating a Store for Production

The steps below outline how to activate your production account so that you can process production transactions.

1. Obtain your activation letter/fax from Moneris.
2. Go to https://www.moneris.com/activate as instructed in the letter/fax.
3. Input your store ID and merchant ID from the letter/fax and click **Activate**.
4. Follow the on-screen instructions to create an administrator account. This account will grant you access to the Merchant Resource Center.
5. Log into the Merchant Resource Center at https://www3.moneris.com/mpg using the user credentials created in step 7.1.
6. Proceed to **ADMIN** and then **STORE SETTINGS**.
7. Locate the API token at the top of the page. Use this API Token along with the store ID that you received in your letter/fax and to send any production transactions through the API.

For more information about how to use the Merchant Resource Center, see the Moneris Gateway Merchant Resource Center User's Guide, which is available at https://developer.moneris.com.

Appendix A Definitions of Required Fields

Table 1: Definitions of Required Fields – Basic and Mag Swipe Transactions

Variable Name	Size/Type	Description
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number pan	*String* 20-character numeric	Credit Card Number with 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 of future expansion and/or potential support of private label card ranges.
Customer ID cust_id (optional)	*String* 50-character alphanumeric	This is an optional field that can be sent as part of a Purchase or Pre-Authorization request. It is searchable from theMerchant Resource Center. It is commonly used for policy number, membership number, student ID or invoice number.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	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 -

Variable Name	Size/Type	Description
		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
Expiry date expdate	*String* 4-character numeric	Expiry Date - format YYMM no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
Order ID order_id	*String* 50-character alphanumeric	Merchant defined unique transaction identifier — must be unique for every purchase, preauth and ind_refund attempt. For refund,completion and purchasecorrection, the order_id must reference the original transaction. Characters allowed: a-z A-Z 0-9 _ - : . @ spaces **NOTE:** This variable also has field definition information exclusive to Vault. For more information, see the table Definitions of Required Fields - Vault Transactions.
POS Code pos_code	*String* 2-character numeric	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

Variable Name	Size/Type	Description
		code.
Track 2 track2	*String*	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.
Transaction number txn_number	*String* 255-character alphanumeric	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 Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.

A.1 Definitions of Request Fields – Credential on File

Credential on File Request Fields

XML Parent Tag

`<cof_info>`

Credential on File request fields below are children of the `<cof_info>` tag.

Variable and Field Name	Type and Limits	Description
Issuer ID `issuer_id` **NOTE:** This variable is required for all merchant-intiated transactions following the first one; upon sending the first transaction, the Issuer ID value is	*String* 15-character numeric	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first

Variable and Field Name	Type and Limits	Description
received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).		time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator `payment_indicator`	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information `payment_information`	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values are: 2 - subsequent transactions (using previously stored payment details)

7.2 Definitions of Request Fields – Level 2

Level 2 field information is contained in the object `<addendum1>`

Table 1: Addendum 1 Request Fields – Level 2 Details

Variable and Field Name	Type and Limits	Description
Customer code `customer_code`	*String* 40-character alphanumeric	A control number, such as purchase order number, project number, department allocation number or name supplied by the customer
Discount amount `discount_amount`	*String*	Monetary amount of discount on the transaction

Variable and Field Name	Type and Limits	Description
	9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	
Freight amount `freight_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Shipping cost on the transaction
Duty amount `duty_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Applicable amount of duty charged on the transaction
National tax amount `national_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of federal-level taxes charged on the transaction, e.g., GST or HST
Other tax amount `other_tax_amount`	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal	Provincial tax amount charged on the transaction, e.g., PST or QST

Variable and Field Name	Type and Limits	Description
	point **EXAMPLE:** 123456.78	
Merchant VAT number `merchant_vat_no`	*String* 15-character alphanumeric	Your GST registration number
Customer VAT number `customer_vat_no`	*String* 13-character alphanumeric	The customer's GST registration number
Destination postal code `destination_zip`	*String* 10-character alphanumeric	Postal code of the destination of goods
Ship from postal code `ship_from_zip`	*String* 10-character alphanumeric	Postal code of the shipping location

7.3 Definitions of Request Fields – Level 3

Level 3 field information is contained in the object `<addendum2>`, an optional nested request object for Level 2/3 transactions.

NOTE: For Level 3 fields sent over the batch method, there is a limit of 98 items per transaction.

Table 1: Addendum 2 Request Fields – Level 3 Details

Variable and Field Name	Type and Limits	Description
Item description `item_description`	*String* 40-character alphanumeric	Description of item; length allowed may differ by card brand
Product code `product_code`	*String* 12-character alphanumeric	UPC or product code of an item
Quantity	*String*	Quantity of units of a particular item being purchased

Variable and Field Name	Type and Limits	Description
`quantity`	9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	
Unit cost `unit_cost`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Unit cost of an item
Total cost `ext_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Total cost of a particular item being purchased (quantity X unit cost)
Unit of measure `uom`	*String* 12-character alphanumeric	Unit of measure used for the item being purchased, as in common use for international trade
Item discount amount `item_discount_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of discount applied to total cost of item

Variable and Field Name	Type and Limits	Description
Item tax amount `item_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of national-level tax charged on item; for Canada, this is GST or HST.
Item other tax amount `item_other_tax_amount`	*String* 9-character decimal; Up to 6 digits (dollars) + decimal point + 2 digits (cents) after the decimal point **EXAMPLE:** 123456.78	Dollar amount of any other taxes applied to the item purchased
Item other tax type `item_other_tax_type`	*String* 4-character alphanumeric	Indicates the type of other tax being applied Possible values: PST - Provincial Sales Tax QST - Quebec Sales Tax (or other)

Appendix B Definitions of Response Fields

Variable Name	Size/Type	Description
AuthCode	8-character numeric	Authorization code returned from the issuing institution
CardType	2-character alphanumeric	Card Type M = MasterCard V = Visa AX = American Express DC = Diners Card NO = Novus / Discover SE = Sears
Complete	true/false	Transaction was sent to authorization host and a response was received
IssuerId	50-character numeric	Credential on File response corresponding to the Issuer ID request variable
Message	100-character alphanumeric	Response description returned from issuing institution.
ReceiptId	50-character alphanumeric	order_id specified in request
RecurSuccess	true/false	Indicates whether the transaction successfully registered
ReferenceNum	18-character numeric	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.

Variable Name	Size/Type	Description
		EXAMPLE: The following illustrates the breakdown of this field where "640123450010690030" is the reference number returned in the message, "64012345" is the terminal id, "001" is the shift number, "069" is the batch number and "003" is the transaction number within the batch. Moneris Host Transaction identifier.
ResponseCode	3-character numeric	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 on the Moneris Developer Portal at https://developer.moneris.com
Ticket	n/a	reserved
TimedOut	true/false	Transaction failed due to a process timing out
TransAmount		
TransDate	yyyy-mm-dd	Processing host date stamp
TransID	20-character alphanumeric	Gateway Transaction identifier
TransTime	##:##:##	Processing host time stamp (24 hour clock)
TransType	2-character numeric	Type of transaction that was performed

Variable Name	Size/Type	Description
		00 = Purchase, Purchase with Vault 01 = Pre-Authorization, Pre-Authorization with Vault 02 = Pre-Authorization Completion 04 = Refund, Independent Refund, Vault CC 11 = Purchase Correction

B.1 Definitions of Response Fields – Vault

Table 1: Definitions of Response Fields – Vault Transactions

Variable Name	Size/Type	Description
ReceiptId	50-character alphanumeric	order_id specified in request
[reserved]	n/a	Future use
ResponseCode	3-character numeric	Transaction Response Code < 50: Transaction approved >= 50: Transaction declined Vault Admin Responses: 001 Successfully registered CC details. Successfully deleted CC details. 983 Can not find previous 986 Incomplete: timed out 987 Invalid transaction Null Error: Malformed XML
TransTime	##:##:##	Processing host time stamp (24 hour clock)
TransDate	yyyy-mm-dd	Processing host date stamp
Complete	true/false	Transaction was sent to authorization host and a response was received
TimedOut	true/false	Transaction failed due to a process timing out

Variable Name	Size/Type	Description
DataKey	50-character alphanumeric	The data_key specified in request. Or, in the case of a res_add_cc transaction it will specify the data_key created by Moneris Solutions.
payment_type	3-character alphanumeric	Indicates the payment type associated with the Vault profile. Possible values: cc – credit card profile
cust_id	50-character alphanumeric	The value stored in the vault, it will be blank if no data is stored.
phone	30-character alphanumeric	The value stored in the vault, it will be blank if no data is stored.
email	30-character alphanumeric	The value stored in the vault, it will be blank if no data is stored.
note	30-character alphanumeric	The value stored in the vault, it will be blank if no data is stored.
pan	alphanumeric	The first 4 and last 4 digits of the credit card number
exp_date	4-character numeric	The credit card expiry date
crypt_type	1-character alphanumeric	The e-commerce indicator

B.2 Definitions of Response Fields – Level 2/3

Table 1: Definitions of Response Fields – Level 2/3 Transactions

Variable Name	Size/Type	Description
AuthCode	8-character alphanumeric	Authorization code returned from the issuing institution

Variable Name	Size/Type	Description
BankTotals	n/a	reserved for future use
CardType	2-character alphanumeric	Credit Card Type M = MasterCard V = Visa AX = American Express
Complete	true/false	Transaction was sent to authorization host and a response was received
CorporateCard	true/false	States whether the card is corporate or not
ISO	2-character numeric	ISO response code
Message	100-character alphanumeric	Response description returned from issuing institution
MessageId	20-character alphanumeric	Processing Host indicator. Reserved.
orig_request	echo	Everything contained in request root of request batch file is echoed back. All fields except pan follows the same format as shown in the request fields definition. pan field will be masked and you will receive only first 4 and last 4 digits of the card.
ReceiptId	50-character alphanumeric	order_id specified in request
ReferenceNum	18-character numeric	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

Variable Name	Size/Type	Description
		batch number and "003" is the transaction number within the batch. Moneris Host Transaction identifier.
ResponseCode	3-character numeric	Transaction Response Code < 50: Transaction approved >= 50: Transaction declined NULL: Transaction was not sent for authorization **NOTE:** If you would like further details on the response codes that are returned please see the Response Codes document available at https://developer-.moneris.com
Ticket	n/a	reserved for future use
TimedOut	true/false	Transaction failed due to a process timing out
TransAmount	14-character decimal	Amount the transaction is processed for
TransDate	yyyy-mm-dd	Processing host date stamp
TransID	20-character alphanumeric	Gateway Transaction identifier. This field is referred in follow-on transaction as txn_number.
TransTime	##:##:##	Processing host time stamp
TransType	alphanumeric	Type of transaction that was performed 00 = Purchase 01 = Preauthorization 02 = Forcepost / Capture 04 = Refund, Independent Refund 11 = Void

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

Appendix D Complete Batch Upload Request - Example

The following sample code illustrates a complete batch upload transaction request, including Vault and Mag Swipe transactions.

```
<?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>

<refund>
<order_id>need_order_id_from_purchasepreauth</order_id>
<amount>2.00</amount>
<txn_number>4678906-04-01</txn_number>
<crypt_type>1</crypt_type>
</refund>

<preauth>
<order_id>Moneris_test_june20_2</order_id>
<amount>15.00</amount>
<pan>5454545454545454</pan>
<expdate>0702</expdate>
<crypt_type>2</crypt_type>
<cust_id>some_cust_id</cust_id>
<cust_info>
<billing>
<first_name>Mary</first_name>
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</billing>
<shipping>
<first_name>Bob</first_name>
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</shipping>
<email>mary@smith.com</email>
<instructions>Please package in wrapping paper</instructions>
<item>
<name>Blue Shoes</name>
<quantity>2</quantity>
<product_code>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>

<ind_refund>
<order_id>need_unique_order_id</order_id>
<amount>80.00</amount>
<pan>4242424242424242</pan>
<expdate>0609</expdate>
<crypt_type>1</crypt_type>
<cust_id>my_cust_id</cust_id>
</ind_refund>

<res_add_cc>
<order_id>Moneris_test_Dec20_2</order_id>
<pan>5454545454545454</pan>
<expdate>1201</expdate>
<crypt_type>7</crypt_type>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
</res_add_cc>

<res_purchase_cc>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<order_id>Moneris_test_Dec20_6</order_id>
<cust_id>My_Customer_Name</cust_id>
<amount>55.00</amount>
<crypt_type>7</crypt_type>
<cust_info>
<billing>
<first_name>Bob</first_name>
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</billing>
<shipping>
<first_name>Bob</first_name>
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
```

```
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</shipping>
<email>bob@smith.com</email>
<instructions>Please deliver to back door</instructions>
<item>
<name>Red Shoes</name>
<quantity>1</quantity>
<product_code>shoes_101</product_code>
<extended_amount>1.00</extended_amount>
</item>
</cust_info>
<recur>
<recur_unit>month</recur_unit>
<start_now>true</start_now>
<start_date>2011/12/01</start_date>
<num_recurs>1</num_recurs>
<period>1</period>
<recur_amount>1.00</recur_amount>
</recur>
</res_purchase_cc>

<res_update_cc>
<order_id>Moneris_test_Dec20_3</order_id>
<data_key>XKwBrc6n8sHIJftK1mNf0TFb3</data_key>
<expdate>1212</expdate>
</res_update_cc>
<res_update_cc>
<order_id>Moneris_test_Dec20_4</order_id>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<pan>5454545454545454</pan>
<expdate>1201</expdate>
<crypt_type>7</crypt_type>
<cust_id>My_Customer_Name</cust_id>
<phone>555-800-1122</phone>
<email>my_email@mail.com</email>
<note>I have no note at this time</note>
</res_update_cc>

<res_preauth_cc>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<order_id>Moneris_test_Dec20_7</order_id>
<cust_id>My_Customer_Name</cust_id>
<amount>55.00</amount>
<crypt_type>7</crypt_type>
<cust_info>
<billing>
<first_name>Bob</first_name>
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</billing>
<shipping>
<first_name>Bob</first_name>
```

```
<last_name>Smith</last_name>
<company_name>None</company_name>
<address>101 Main St</address>
<city>Springfield</city>
<province>YU</province>
<postal_code>Z1Z1Z1</postal_code>
<country>CAD</country>
<phone_number>919-555-5555</phone_number>
<fax>919-555-5550</fax>
<tax1>0.10</tax1>
<tax2>0.20</tax2>
<tax3>0.00</tax3>
<shipping_cost>0.00</shipping_cost>
</shipping>
<email>bob@smith.com</email>
<instructions>Please deliver to back door</instructions>
<item>
<name>Red Shoes</name>
<quantity>1</quantity>
<product_code>shoes_101</product_code>
<extended_amount>1.00</extended_amount>
</item>
</cust_info>
</res_preauth_cc>

<res_ind_refund_cc>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
<order_id>Moneris_test_Dec20_8</order_id>
<cust_id>My_Customer_Name</cust_id>
<amount>55.00</amount>
<crypt_type>7</crypt_type>
</res_ind_refund_cc>

<res_delete>
<order_id>Moneris_test_Dec20_5</order_id>
<data_key>RbFbmySFugYNRSmTaN6Gkm08r</data_key>
</res_delete>

<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>
```