Moneris™

BE PAYMENT READY

# Batch Upload Merchant Integration Guide

.csv

Canada only

Version 1.2.0 - May 2018

Copyright © Moneris Solutions, 2017

All rights reserved. No part of this publication may be reproduced, stored in retrieval systems, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Moneris Solutions Corporation.

Moneris Gateway

Copyright Notice

Copyright © 2016 Moneris Solutions, 3300 Bloor Street West, Toronto, Ontario, M8X 2X2

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

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

Trademarks

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

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

Printed in Canada.

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

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

# 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:<br>clientintegrations@moneris.com<br>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:<br>1-866-319-7450<br>eproducts@moneris.com<br>Hours: 8am to 8pm ET | If your application is already live and you need production support, contact Moneris Customer Service:<br>onlinepayments@moneris.com<br>1-866-319-7450<br>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:

1. Have SFTP client software which must use SSH2
2. Create the Batch File which must have the ".csv" extension

Additional requirements for VAU and ABU updating:

1. Create the VAU Batch request .csv File
2. Create the ABU Batch request .csvFile
3. Create and use GPG public and private key (encryption)

# 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 .csv Batch File option refers to the ability to upload batch files where the fields are presented in a comma delimited 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 .csv 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 about4.1 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

### 1.1.1 Implementing VAU and ABU - Summary of Process

The process of batch uploads for VAU and ABU has additional requirements. You will need to follow these steps in the test environment before moving to production with VAU or ABU:

1. Request a VAU/ABU test SFTP account
2. Create a GPG public and private key.
3. Upload the GPG public key to your SFTP home directory
4. Upload your test VAU batch request file to your SFTP home directory
5. Upload your test ABU batch request file to your SFTP home directory
6. Download the test VAU response file
7. Download the test ABU response file
8. Decrypt the files

Once the above is done and approved by Moneris, you will need to follow the same process in production when processing real files.

**NOTE:**

To get your test and/or production SFTP username and password, please contact Moneriscustomer 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.

### 1.1.1.1 GPG Keys

The VAU and ABU response files will be encrypted using the GPG public key method. Below are the required steps. Please refer to Appendix - Generating a Key Using GPG4win for an example.

**NOTE:** For PGP users, the merchant ID will have to be added in the Full name field

1. Merchant will create public/private key pair with encryption software that will create a “comments” field when the public key is generated. The comment filed is a prompt when creating the key.
2. The merchant must place the merchant ID into the comment field. For the test environment, please use your production merchant ID. Otherwise, your response files will not be generated.
3. Merchant will export the public key in ASCII key file format and rename the file name extension .gpg (e.g., merchant1.gpg)
4. Merchant SFTP's the public key into merchant's SFTP home directory on Moneris server
5. Moneris will use the public key to encrypt the response file
6. Any new GPG public key uploaded to the Moneris server will replace the previous public key and will be used accordingly

**NOTE:** Only the response file will be encrypted. The request file should not be encrypted.

**NOTE:** When the public encryption key is uploaded into the merchant’s SFTP Home Directory on the MonerisServer, it will disappear after a few seconds. At this point, the merchant-created public key is updated.

# 2 Transaction Types and Transaction Process Flows

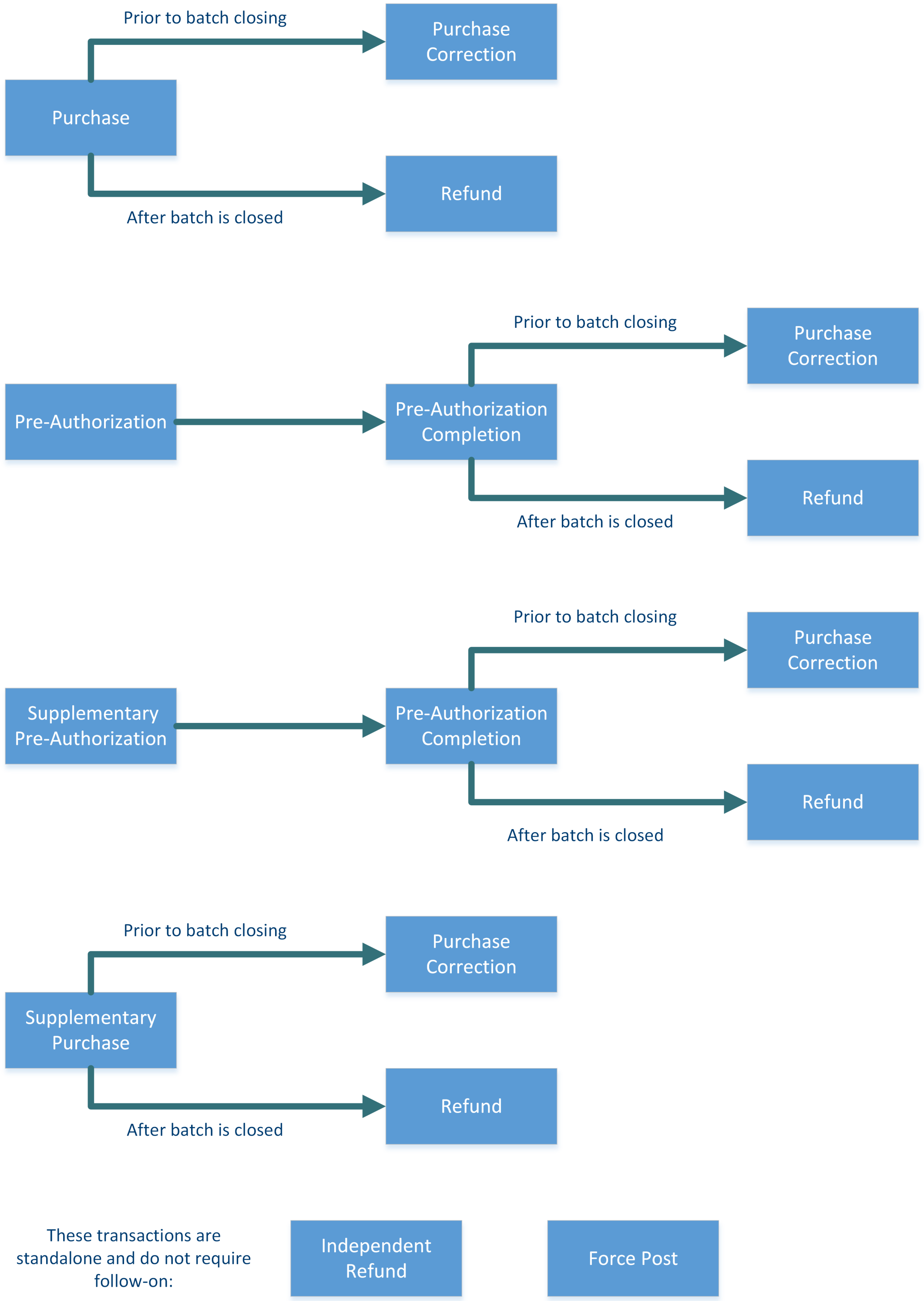
- 2.1 Basic Transactions
- 2.2 Mag Swipe Transactions
- 2.3 Vault Transactions
- 2.6 Visa Account Updater and Automatic Billing Updater

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
- 2.1.3 Pre-Authorization
- 2.1.4 Pre-Authorization Completion
- 2.1.5 Force Post
- 2.1.6 Purchase Correction
- 2.1.7 Refund
- 2.1.8 Independent Refund

### 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 – See 2.1.2.2 Supplementary Purchase
Recurring Billing – See below
Recurring Billing and Credential on File - See 2.1.2.5 Purchase with Recurring Billing and Credential on File
Credential on File - See 2.1.3.3 Purchase with Credential on File
Customer ID and Credential on File – See 2.1.2.4 Supplementary Purchase with Credential on File

### CSV Request Format – Purchase

```
purchase, order_id, amount, pan, exp_date, crypt_type
```

### CSV Request Format – Purchase with Recurring Billing (optional)

```
purchase, order_id, amount, pan, exp_date, crypt_type, recur_unit, start_now,
start_date, num_recurs, period, recur_amount
```

### CSV Response Format – Purchase

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Purchase

**Table 1: Required Fields - Purchase transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99<br>**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<br>`pan` | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-Commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Trans- |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | action (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

### Optional Transaction Request Variables – Purchase

**Table 2: Optional Request Fields - Purchase transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Customer ID<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |
| Credential on File Info<br>`cof_info` | *Object* | Required for transactions using stored cardholder credentials<br>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. |

### Recurring Billing Transactions Request Fields

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

**Table 3: Required Fields - Recurring Billing Transactions**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| recur_unit | *String*<br>day, week, month, eom | The unit that you wish to use as a basis for the Interval. This can |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | 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*<br>0 – 999 characters, numeric | This is the number of recur_ units you wish to pass between billing cycles.<br>Example :<br>period = 3, recur_unit=month -> Card will be billed every 3 months.<br>period = 4, recur_unit=weeks -> Card will be billed every 4 weeks.<br>period = 45, recur_unit=day -> Card will be billed every 45 days.<br>Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future. |
| start_date | *String*<br>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*<br>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 dif- |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | fer from the recur amount billed on a regular basis thereafter. |
| recur_amount | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:**<br>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 9999999.99. This is the amount that will be billed on the start_date and every interval thereafter. |
| num_recurs | *String*<br>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 1 Customer Information (cust_info) Fields.

## 2.1.2.1 Purchase Transactions With Additional Features

The positional nature of CSV requires separate transaction types for adding additional features.

These additional Purchase transaction types include:

**Supplementary Purchase** – Adds the Customer ID field to the Purchase
**Purchase with Credential on File** – Adds Credential on File information fields to the Purchase
**Supplementary Purchase with Credential on File** – Adds both Customer ID and Credential on File Information fields to the Purchase

## 2.1.2.2 Supplementary Purchase

Supplementary Purchase is a Purchase transaction with an optional Customer ID field included. The request fields are the same as a basic Purchase with the addition of the Customer ID.

### CSV Request Format –

```
purchase_supp, order_id, amount, pan, exp_date, crypt_type, cust_id
```

### CSV Response Format –

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### 2.1.2.3 Purchase with Credential on File

Purchase with Credential on File is the Purchase transaction with the Credential on File fields included.

### CSV Request Format – Purchase with Credential on File

```
purchase_cof, order_id, amount, pan, exp_date, crypt_type, issuer_id, payment_
indicator, payment_information
```

### CSV Response Format – Purchase with Credential on File

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, IssuerId
```

### 2.1.2.4 Supplementary Purchase with Credential on File

Supplementary Purchase with Credential on File is used for sending a Purchase transaction along with both the Customer ID and Credential on File information.

Supplementary Purchase with Credential on File has the same request fields as Purchase transactions, plus Customer ID and the Credential on File required fields.

### CSV Request Format – Supplementary Purchase with Credential on File

```
purchase_supp_cof, order_id, amount, pan, exp_date, crypt_type, cust_id,
issuer_id, payment_indicator, payment_information
```

### CSV Response Format – Supplementary Purchase with Credential on File

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, IssuerId
```

### 2.1.2.5 Purchase with Recurring Billing and Credential on File

Purchase with Recurring Billing and Credential on File is the Purchase transaction with the Recurring Billing fields and the Credential on File fields included.

### CSV Request Format – Purchase with Recurring Billing and Credential on File

```
purchase_recur_cof, order_id, amount, pan, exp_date, crypt_type, recur_unit,
start_now, start_date, num_recurs, period, recur_amount, issuer_id, payment_
indicator, payment_information
```

### CSV Response Format – Purchase with Recurring Billing and Credential on File

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, IssuerId
```

## 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 ID - See 2.1.3.2 Supplementary Pre-Authorization
Credential on File - See 1 Pre-Authorization with Credential on File
Customer ID and Credential on File - See 2.1.3.4 Supplementary Pre-Authorization with Credentials on File

### CSV Request Format – Pre-Authorization

```
preauth, order_id, amount, pan, exp_date, crypt_type
```

### CSV Response Format – Pre-Authorization

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Pre-Authorization

**Table 1: Required Fields – Pre-Authorization transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>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<br>`pan` | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV) |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | 7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

## Optional Transaction Request Variables – Pre-Authorization

**Table 2: Optional Request Fields - Pre-Authorization transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Customer ID<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |
| Customer Information<br>`cust_info` | *Object*<br>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. |
| Credential on File Info<br>`cof_info` | *Object* | Required for transactions using stored cardholder credentials.<br>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. |

### Customer Information Request Fields

For information about the Customer Information object fields, see 1 Customer Information (cust_ info) Fields.

#### 2.1.3.1 Pre-Authorization Transactions With Additional Features

The positional nature of CSV requires separate transaction types for adding additional features.

These additional Pre-Authorization transaction types include:

**Supplementary Pre-Authorization** – Adds the Customer ID field to the Pre-Authorization
**Pre-Authorization with Credential on File** – Adds Credential on File information fields to the Pre-Authorization
**Supplementary Pre-Authorization with Credential on File** – Adds both Customer ID and Credential on File Information fields to the Pre-Authorization

#### 2.1.3.2 Supplementary Pre-Authorization

Supplementary Pre-Authorization is a Pre-Authorization transaction with an optional Customer ID field included.

**CSV Request Format – Supplementary Pre-Authorization**

```
preauth_supp, order_id, amount, pan, exp_date, crypt_type, cust_id
```

**CSV Response Format – Supplementary Pre-Authorization**

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

#### 2.1.3.3 Purchase with Credential on File

Purchase with Credential on File is the Purchase transaction with the Credential on File fields included.

**CSV Request Format – Purchase with Credential on File**

```
purchase_cof, order_id, amount, pan, exp_date, crypt_type, issuer_id, payment_
indicator, payment_information
```

**CSV Response Format – Purchase with Credential on File**

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, IssuerId
```

#### 2.1.3.4 Supplementary Pre-Authorization with Credentials on File

Supplementary Pre-Authorization with Credential on File has the same request fields as the basic Pre-Authorization transaction, plus Customer ID and the Credential on File required fields.

### CSV Request Format – Supplementary Pre-Authorization with Credentials on File

```
preauth_supp_cof, order_id, amount, pan, exp_date, crypt_type, cust_id,
issuer_id, payment_indicator, payment_information
```

### CSV Response Format – Supplementary Pre-Authorization with Credentials on File

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, IssuerId
```

## 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.

### CSV Request Format – Pre-Authorization Completion

```
completion, order_id (from preauth), amount, txn_number, crypt_type
```

### CSV Response Format – Pre-Authorization Completion

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Pre-Authorization Completion

**Table 1: Required Fields – Pre-Authorization Completion transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99 |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | **EXAMPLE:**<br>123456.78 | |
| Transaction number<br>`txn_number` | *String*<br>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.<br>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<br>`crypt_type` | *String*<br>1-character alphanumeric | E-Commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>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

### Optional Features

Customer ID

### CSV Request Format – Force Post

```
forcepost, order_id, amount, pan, exp_date, auth_code, crypt_type
```

### CSV Response Format – Force Post

```
order_id, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate, Com-
plete, Message,amount, CardType, TxnNumber, TimedOut, reserved, reserved
```

**Table 1: Required Fields – Force Post transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>order_id | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>amount | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>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<br>pan | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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 poten- |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | tial support of private label card ranges. |
| Expiry Date<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| Authorization code<br>`auth_code` | *String*<br>8-character alphanumeric | Authorization code provided in the transaction response from the issuing bank |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>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<br>`cust_id` | *String*<br>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.<br>Can be searched from the MonerisMerchant 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.

### CSV Request Format – Purchase Correction

```
purchasecorrection, order_id, txn_number, crypt_type
```

### CSV Response Format – Purchase Correction

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Purchase Correction

**Table 1: Required Fields – Purchase Correction transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Transaction number<br>`txn_number` | *String*<br>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. |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | 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<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Installment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>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.

### CSV Request Format – Refund

```
refund, order_id, amount, txn_number, crypt_type
```

### CSV Response Format – Refund

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Refund

**Table 1: Required Fields – Refund transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:**<br>123456.78 | Amount of the transaction.<br>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<br>`txn_number` | *String*<br>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.<br>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<br>`crypt_type` | *String*<br>1-character alphanumeric | E-Commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | 3 - Mail Order / Telephone Order - Installment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>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.

### Optional Features

Customer ID - See Optional Fields table below

### CSV Request Format – Independent Refund

```
ind_refund, order_id, amount, pan, exp_date, crypt_type
```

### CSV Response Format – Independent Refund

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket
```

### Transaction Request Variables – Independent Refund

**Table 1: Required Fields – Independent Refund transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>50-character alphanumeric | Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | transactions of these types may have the same order ID.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>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<br>`pan` | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>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<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant 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 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, exp_date, pos_code |
| Mag Swipe Pre-Authorization (track2_preauth) | track2_preauth, order_id, cust_id, pan, exp_date, 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 |

| Transaction Type | Fields |
|---|---|
|  | track2_completion) |
| Mag Swipe Independent Refund<br>(track2_ind_refund) | track2_ind_refund, order_id, cust_id, amount, track2, pan, exp_date, pos_code |

### 2.2.1.1 Example - Mag Swipe Transaction

The following sample code illustrates a Mag Swipe transaction request. A corresponding example for the transaction response follows.

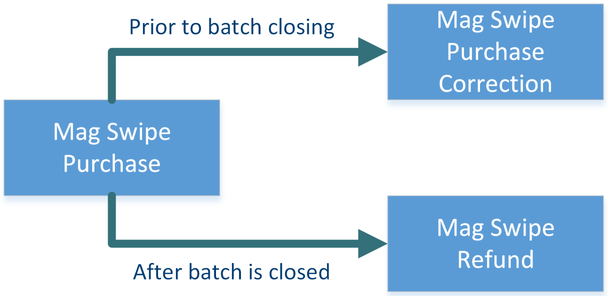
> **NOTE:** In a Mag Swipe/track2 transaction, to preserve the order of the fields, you must still include a comma delimiter if you omit ‘pan’ or ‘exp_date’ fields. This may be seen in examples 3 and 4, where the ‘pan’ and ‘exp_date’ fields have been omitted but their positions have still been marked.

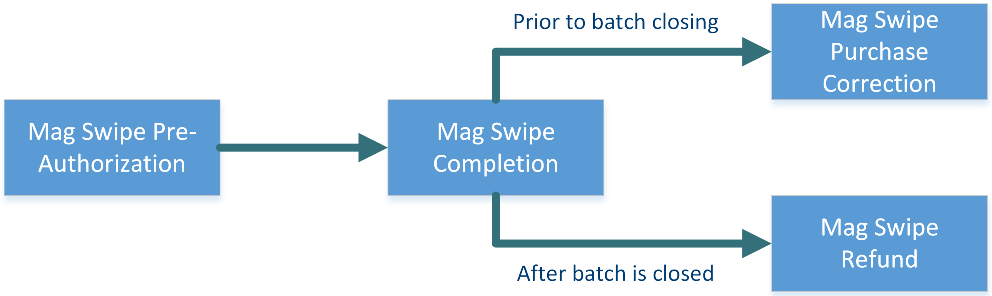
```
purchase,order_1_testing,13.00,4242424242424242,0304,1
purchase_supp,order_2_testing,2.00,4242424242424242,0908,1,customer_1
purchase,order_3_testing,13.00,5454545454545454,0403,1
preauth,order_4_testing,14.00,4242424242424242,0503,1
track2_purchase,track2_testing_1,cust_id,1.00,;5258984987184986=06061016091001060602?,,,00
track2_preauth,track2_testing_2,cust_id,9.00,;5258984987184986=06061016091001060602?,,,00
track2_ind_refund,track2_testing_3,cust_id,8.00,;5258984987184986=06061016091001060602?,,,00
```

### Corresponding Example Response for Mag Swipe Transaction

```
order_1_testing,660021810013368320,027,01,010460,11:04:27,2006-06-20,00,true,APPROVED *
    =,13.00,V,63790-832-0,false,,null
order_2_testing,660021810013368330,027,01,009494,11:04:28,2006-06-20,00,true,APPROVED *
    =,2.00,V,63791-833-0,false,,null
order_3_testing,660021810013368340,027,01,010461,11:04:30,2006-06-20,00,true,APPROVED *
    =,13.00,M,63792-834-0,false,,null
order_4_testing,660021810013368350,027,01,010462,11:04:31,2006-06-20,01,true,APPROVED *
    =,14.00,V,63793-835-0,false,,null
track2_testing_1,660021810013842170,027,01,008348,13:24:34,2006-11-24,00,true,APPROVED *
    =,1.00,M,97547-217-0,false,,null
track2_testing_2,660021810013842180,027,01,007323,13:24:35,2006-11-24,01,true,APPROVED *
    =,9.00,M,97548-218-0,false,,null
track2_testing_3,660021810013842190,027,01,000000,13:24:35,2006-11-24,04,true,APPROVED *
    =,8.00,M,97549-219-0,false,,null
```

### 2.2.2 Mag Swipe Transactions Process Flow





These transactions are standalone and do not require follow-on:

Mag Swipe Independent Refund

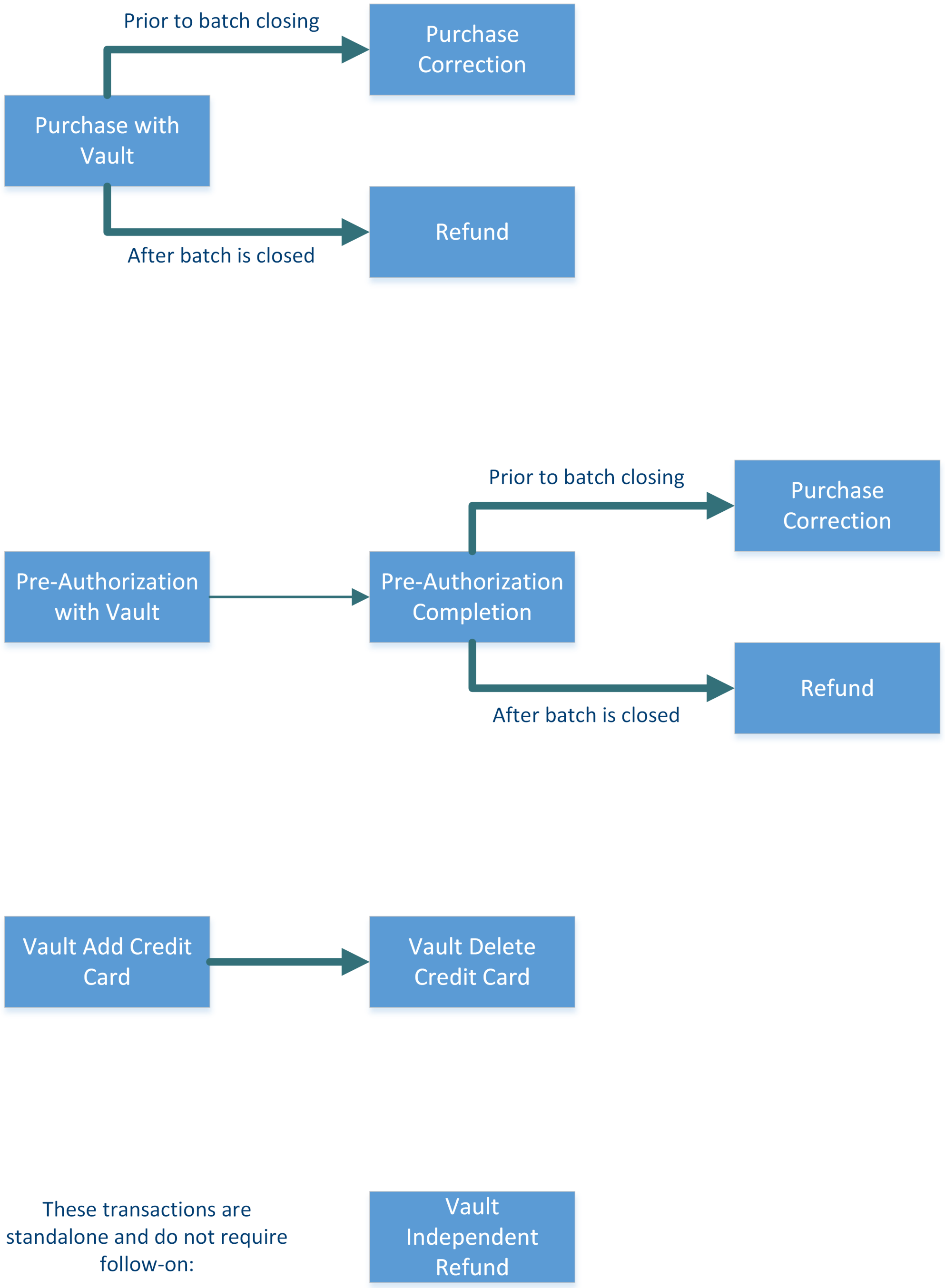
## 2.3 Vault Transactions

- 2.3.1 Vault Transactions Process Flow
- 2.3.2 Vault Administrative Transactions
- 2.3.3 Vault Financial 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 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.

#### Optional Features

Customer ID - See Optional Fields table below

#### CSV Request Format – Vault Add Credit Card

```
res_add_cc, order_id, cust_id,phone, email, note, pan, exp_date, crypt_type
```

#### CSV Response Format – Vault Add Credit Card

```
order_id, reserved, ResponseCode, reserved, reserved, TransTime, TransDate,
reserved, Complete, Message, reserved, reserved, reserved, TimedOut, reserved,
reserved, data_key, payment_type, cust_id, phone, email, note, pan, exp_date,
crypt_type, reserved, reserved, reserved, reserved
```

#### 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<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Credit card number<br>`pan` | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

### 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<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |
| Phone Number<br>`phone` | *String*<br>30-character alphanumeric | Phone number of the customer |
| Note<br>`note` | *String*<br>30-character alphanumeric | Used for supplementary information |
| Email address<br>`email` | *String*<br>30-character alphanumeric | Email address of the customer |

## 2.3.2.2 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.

### Optional Features

Customer ID - See Optional Fields table below

### CSV Request Format – Vault Update Credit Card

```
res_update_cc,order_id,cust_id,phone,email,note,pan,exp_date,crypt_type
```

### CSV Response Format – Vault Update Credit Card

```
order_id, reserved, ResponseCode, reserved, reserved, TransTime, TransDate,
reserved, Complete, Message, reserved, reserved, reserved, TimedOut, reserved,
reserved, data_key, payment_type, cust_id, phone, email, note, pan, exp_date,
crypt_type, reserved, reserved, reserved, reserved
```

## 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<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Data key<br>`data_key` | *String*<br>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<br>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 |
|---|---|---|
| Credit card number<br>`pan` | *String*<br>20-character numeric | Credit Card Number with no spaces or dashes.<br>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<br>`exp_date` | *String*<br>4-character numeric<br>YYMM format | Expiry date with no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| E-commerce indicator<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |
| Customer ID<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Phone Number<br>`phone` | *String*<br>30-character alphanumeric | Phone number of the customer |
| Note<br>`note` | *String*<br>30-character alphanumeric | Used for supplementary information |
| Email address<br>`email` | *String*<br>30-character alphanumeric | Email address of the customer |

### 2.3.2.3 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.

#### CSV Request Format – Vault Delete

```
res_delete,order_id,data_key
```

#### CSV Response Format – Vault Delete

```
order_id, reserved, ResponseCode, reserved, reserved, TransTime, TransDate,
reserved, Complete, Message, reserved, reserved, reserved, TimedOut, reserved,
reserved, data_key, payment_type, cust_id, phone, email, note, pan, exp_date,
crypt_type, reserved, reserved, reserved, reserved
```

#### Transaction Request Variables – Vault Delete

**Table 1: Required Fields – Vault Update Credit Card transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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. |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Data key<br>`data_key` | *String*<br>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<br>The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered |

### 2.3.2.4 Example - Vault Admin Transaction

```
res_add_cc,mjr-res-add-210510-251-1,moneris,phone,email,note,4242424242424242,1111,7
res_delete,mjr-res-delete-210510-251-1,E08TCqdXsiOhDey9YZDa7QDMa
```

### Corresponding Example Response -for Vault Admin Transaction

```
mjr-res-add-210510-251-1,null,001,null,null,14:52:49,2010-05-21,null,true,Successfully registered CC
    details.,null,null,null,false,,,we3u5tak7ce4UPiy1xRnDqqju,cc,moneris,phone,email,note,4242***42
    42,1111,7,,,,
mjr-res-delete-210510-251-1,null,001,null,null,14:53:13,2010-05-21,null,true,Successfully deleted CC
    details.,null,null,null,false,,,E08TCqdXsiOhDey9YZDa7QDMa,cc,moneris,phone,email,note,4242***42
    42,1111,7,,,,
```

## 2.3.3 Vault Financial Transactions

- 2.3.3.1 Purchase with Vault
- 2.3.3.3 Pre-Authorization with Vault
- 2.3.3.4 Vault Independent Refund

### 2.3.3.1 Purchase with Vault

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

Customer ID - see Optional Fields table below
Recurring Billing
Credential on File

#### CSV Request Format – Purchase with Vault

```
res_purchase_cc, order_id, cust_id, amount, data_key, crypt_type
```

#### CSV Response Format – Purchase with Vault

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, data_key, payment_type, cust_id, phone, email, note, pan,
exp_date, crypt_type, reserved, reserved, reserved, reserved
```

#### Transaction Request Variables – Purchase with Vault

**Table 1: Required Fields – Purchase with Vault transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Data key<br>`data_key` | *String*<br>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<br>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<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

**Table 2: Optional Fields – Purchase with Vault transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Expiry Date<br>`exp_date` | *String*<br>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. |
| Customer ID<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |
| Credential on File Info<br>`cof_info` | *Object* | Required for transactions using stored cardholder credentials<br>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. |

### Recurring Billing Transactions Request Fields

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

**Table 3: Required Fields - Recurring Billing Transactions**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| recur_unit | *String*<br>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*<br>0 – 999 characters, numeric | This is the number of recur_ units you wish to pass between billing cycles.<br>Example : |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | period = 3, recur_unit=month -> Card will be billed every 3 months.<br>period = 4, recur_unit=weeks -> Card will be billed every 4 weeks.<br>period = 45, recur_unit=day -> Card will be billed every 45 days.<br>Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future. |
| start_date | *String*<br>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*<br>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*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point | Amount of the recurring transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99. This is the amount that will be billed on the start_date and every interval there- |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | **EXAMPLE:** 123456.78 | after. |
| num_recurs | *String*<br>1 – 99 characters, numeric | The number of times to recur the transaction. |

| Variable Name | Limits | Description |
|---|---|---|
| Issuer ID<br>`issuer_id`<br>**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*<br>15-character numeric<br>variable length | Unique identifier for the cardholder's stored credentials<br>Sent back in the response from the card brand when processing a transaction<br>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<br>`payment_indicator` | *String*<br>1-character alphabetic | Indicates the intended or current use of the credentials<br>Possible values for subsequent transactions:<br>R - recurring<br>U - unscheduled merchant-initiated transaction<br>Z - unscheduled cardholder-initiated transaction |
| Payment Information<br>`payment_information` | *String*<br>1-character numeric | Describes whether the transaction is the first or subsequent in the series<br>Possible values are: |

| Variable Name | Limits | Description |
|---|---|---|
| | | 2 - subsequent transactions (using previously stored payment details) |

### 2.3.3.2 Purchase with Vault and Credential on File

Purchase with Vault and Credential on File is a Purchase with Vault transaction with the Credential on File fields included. The fields are the same as a Purchase with Vault with the addition of the Credential on File fields.

#### CSV Request Format – Purchase with Vault and Credential on File

```
res_purchase_cc_cof, order_id,cust_id,amount,data_key,crypt_type, issuer_id,
payment_indicator, payment_information
```

#### CSV Response Format – Purchase with Vault and Credential on File

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, data_key, payment_type, cust_id, phone, email, note, pan,
exp_date, crypt_type, reserved, reserved, reserved, reserved, IssuerId
```

### 2.3.3.3 Pre-Authorization with Vault

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 ID – See Optional Fields table below
Credential on File – See 1 Pre-Authorization with Vault and Credential on File

#### CSV Request Format –

```
res_preauth_cc, order_id, cust_id, amount, data_key, crypt_type
```

#### CSV Response Format –

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, data_key, payment_type, cust_id, phone, email, note, pan,
exp_date, crypt_type, reserved, reserved, reserved, reserved
```

#### 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 | *String* | Merchant-defined transaction |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| order_id | 50-character alphanumeric | 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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>amount | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |
| Data key<br>data_key | *String*<br>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<br>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<br>crypt_type | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| | | 3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

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

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Customer ID<br>`cust_id` | *String*<br>50-character alphanumeric | Merchant-defined value, used for additional identification purposes<br>**EXAMPLE:** policy number, membership number, student ID, invoice number.<br>Can be searched from the MonerisMerchant Resource Center |
| Credential on File Info<br>`cof_info` | *Object* | Required for transactions using stored cardholder credentials<br>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. |

## Customer Information Request Fields

For information about the Customer Information object fields, see 1 Customer Information (cust_ info) Fields.

### 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.

#### Optional Features

Customer ID - See Optional Values table below

#### CSV Request Format – Vault Independent Refund

```
res_ind_refund_cc, order_id, cust_id, amount, data_key, crypt_type
```

#### CSV Response Format – Vault Independent Refund

```
ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, TxnNumber, TimedOut,
BankTotals, Ticket, data_key, payment_type, cust_id, phone, email, note, pan,
exp_date, crypt_type, reserved, reserved, reserved, reserved
```

#### Transaction Request Variables – Vault Independent Refund

**Table 1: Required Fields – Vault Independent Refund transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Order ID<br>`order_id` | *String*<br>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.<br>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction. |
| Amount<br>`amount` | *String*<br>9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:**<br>123456.78 | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 9999999.99 |

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Data key<br>`data_key` | *String*<br>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<br>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<br>`crypt_type` | *String*<br>1-character alphanumeric | E-commerce Indicator possible values:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non-secure Transaction (Web or Email Based)<br>9 - SET non-authenticated transaction |

**Table 2: Optional Fields – Vault Independent Refund transaction**

| Variable and Field Name | Type and Limits | Description |
|---|---|---|
| Customer ID<br>`cust_id` | *String*<br>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.<br>Can be searched from the MonerisMerchant Resource Center |

### 2.3.3.5 Example - Vault Financial Transaction

The following sample code illustrates a basic transaction request that also uses Vault financial transaction features, followed by a sample of the corresponding response. For information on the required fields for administrative transactions using Vault, see Sending Vault Administrative Transactions.

```
purchase,order_1_testing,13.00,4242424242424242,0304,1
purchase_supp,order_2_testing,2.00,4242424242424242,0908,1,customer_1
purchase,order_3_testing,13.00,5454545454545454,0403,1
preauth,order_4_testing,14.00,4242424242424242,0503,1
res_purchase_cc,mjr-res-purch-210510-251-1,moneris,1.00,M6eGTseSjlDBxYRICD3rgAhBn,1
res_preauth_cc,mjr-res-preauth-210510-251-1,moneris,1.00,Ef3QO7bzCE3hTzmDqjvC5dMdl,1
res_ind_refund_cc,mjr-res-indrefund-210510-251-1,moneris,1.00,dOeX9Hu7pRsLIJNcBphTTOmer,1
```

### Corresponding Example Response for Vault Financial Transaction

```
order_1_testing,660021810013368320,027,01,010460,11:04:27,2006-06-20,00,true,APPROVED *
    =,13.00,V,63790-832-0,false,,null
order_2_testing,660021810013368330,027,01,009494,11:04:28,2006-06-20,00,true,APPROVED *
    =,2.00,V,63791-833-0,false,,null
order_3_testing,660021810013368340,027,01,010461,11:04:30,2006-06-20,00,true,APPROVED *
    =,13.00,M,63792-834-0,false,,null
order_4_testing,660021810013368350,027,01,010462,11:04:31,2006-06-20,01,true,APPROVED *
    =,14.00,V,63793-835-0,false,,null
mjr-res-purch-210510-251-1,660035500012243850,027,01,073570,14:52:55,2010-05-21,00,true,APPROVED *
    =,1.00,M,112072-0_7,false,,,M6eGTseSjlDBxYRICD3rgAhBn,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
mjr-res-preauth-210510-251-1,660035500012243860,027,01,073571,14:53:01,2010-05-21,01,true,APPROVED *
    =,1.00,M,112073-0_7,false,,,Ef3QO7bzCE3hTzmDqjvC5dMdl,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
mjr-res-indrefund-210510-251-1,660035500012243870,027,01,740332,14:53:08,2010-05-21,04,true,APPROVED
    * =,1.00,M,112074-0_7,false,,,dOeX9Hu7pRsLIJNcBphTTOmer,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
```

## 2.4 Credential on File

- 2.4.1 About Credential on File Transactions – Batch Upload
- 2.4.2 Credential on File Info Object Request Fields

## 2.4.1 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

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.

## 2.4.2 Credential on File Info Object Request Fields

| Variable Name | Limits | Description |
|---|---|---|
| Issuer ID<br>`issuer_id`<br>**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*<br>15-character numeric<br>variable length | Unique identifier for the cardholder's stored credentials<br>Sent back in the response from the card brand when processing a transaction<br>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<br>`payment_indicator` | *String*<br>1-character alphabetic | Indicates the intended or current use of the credentials<br>Possible values for subsequent transactions:<br>R - recurring<br>U - unscheduled merchant-initiated transaction |

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

## 2.5 Recurring Billing

- 2.5.1 About Recurring Billing Transactions – Batch Upload
- 2.5.2 Sending Recurring Transactions
- 2.5.3 Example – Recurring Transactions

### 2.5.1 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)
Purchase with Recurring Billing and Credential on File

### 2.5.2 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.

### 2.5.3 Example – Recurring Transactions

The following sample code illustrates a recurring transaction request using Purchase, followed by a sample of the corresponding response.

```
purchase,order_1_recurring,3.00,4242424242424242,0712,2,week,true,2006/10/30,4,2,1.00
purchase,order_2_recurring,4.00,4242424242424242,0803,2,day,false,2006/10/15,30,10,4.00
purchase,order_3_recurring,5.00,4242424242424242,0610,2,month,true,2006/11/11,12,1,1.00
```

#### Corresponding Example Response - Recurring Transactions

```
order_1_recurring,660021810013368380,027,01,008598,11:28:28,2006-06-20,00,true,APPROVED * =:
    Recurring transaction successfully registered.,3.00,V,63796-838-0,false,,null,true
order_2_recurring,null,null,null,null,null,null,null,true,Recurring transaction successfully
    registered.,null,null,null,false,,null,true
order_3_recurring,660021810013368390,027,01,010535,11:28:34,2006-06-20,00,true,APPROVED * =:
    Recurring transaction successfully registered.,5.00,V,63797-839-0,false,,null,true
```

## 2.6 Visa Account Updater and Automatic Billing Updater

In an effort to alleviate merchants' effort in obtaining account information updates for recurring and card on file transactions, Visa and MasterCard introduced "Visa Account Updater (VAU)" and "Automatic Billing Updater (ABU)", respectively. VAU and ABU support account updates for online, offline and recurring payments. The updates include:

1. Brand conversion
2. Expiry date changes
3. Card Upgrades
4. Re-issued cards due to fraud or new programs
5. Lost or stolen cards

The VAU and ABU products offer merchants a secure mechanism to exchange and to update cardholder account information. These programs support the processing of recurring payment transactions or any card on file program.

Using VAU or ABU, issuers can communicate changes to cardholder account information to Moneris and their participating merchants. These merchants can then quickly and easily update their billing records and significantly reduce the number of authorization request declines.

# 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 .csv 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 59.

## 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 ".csv", 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.

For transaction files using Visa Account Update (VAU) and MasterCard's Automatic Bill Updater (ABU) , the filename you create must also be appended with .vau and .abu as in:

```
file_name.csv.vau
file_name.csv.abu
```

# 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 | https://esqa.moneris.com | https://www3.moneris.com |
| Port | 22 | 22 |
| Server Type | SFTP using SSH2 | SFTP using SSH2 |
| Username | store3 | Provided by Moneris<br>(please refer to NOTE) |
| Password | store3 | Provided by Moneris<br>(please refer to NOTE) |

**NOTE:**

To get your test and/or production SFTP username and password, please contact Moneriscustomer 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:

"Uploading a Batch File and Getting the Response" on page 61

"Moving to Production" on page 68

# 5 Uploading a Batch File and Getting the Response

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.csv | testing_june_20.csv.out |

## 5.1 Filename Conventions for VAU and ABU

**NOTE:** An incremented sequence number will be appended to the filename to differentiate the response files from each other when VAU/ABU processing occurs over several days with large request files.

**EXAMPLE:** Merchant_original_file_name_1.csv.(abu|vau).out, Merchant_original_file_name_2.csv.(abu|vau).out

**Table 1: Filename Conventions for VAU and ABU**

| Program | Request File | Response File |
|---|---|---|
| ABU | file_name.csv.abu | file_name.csv.abu.out |
| VAU | file_name.csv.vau | file_name.csv.vau.out |

## 5.2 What Is Contained in a Transaction Response File?

Within the batch response file, for each transaction sent you will receive a response message. For a full description of each field please refer to "Definitions of Response Fields" on page 72.

After the file is successfully processed go to the /out directory of the SFTP account and download the .out file.  This will contain all the transaction responses.  Please review the response file for all of the transactions.

## 5.2.1 Response Format for Basic Transactions

```
order_id, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,
TransType, Complete, Message, TransAmount, CardType, txn_number, TimedOut,
BankTotals, Ticket
```

**EXAMPLE**

```
    order_1_testing,660021810013368320,027,01,010460,11:04:27,2006-06-
        20,00,true,APPROVED           *                    =,13.00,V,63790-832-
        0,false,,null
```

## 5.2.2 Response Formats for Vault Transactions

When a Vault transaction is successfully processed you will receive a response message made up of two parts.

The first part has an identical response format to all other non-Vault financial transactions (please refer above). The second part will follow directly after part 1 and will include all of the fields registered within the profile. The data that will be included in part 2 depends on what type of profile was registered and what data was initially added.

Any field that is not stored in the profile will still be included in the response by having its position marked by the comma delimiter.

Response Format for Vault Financial Transactions

Part 1

order_id, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate, TransType, Complete, Message, TransAmount, CardType, txn_number, TimedOut, BankTotals, Ticket

Part 2

data_key, payment_type, cust_id, phone, email, note, pan, exp_date, crypt_type[reserved], [reserved], [reserved], [reserved]

**EXAMPLE**

```
mjr-res-purch-210510-251-1,660035500012243850,027,01,073570,14:52:55,2010-05-
    21,00,true,APPROVED * =,1.00,M,112072-0_
    7,false,,,M6eGTseSjlDBxYRICD3rgAhBn,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
mjr-res-preauth-210510-251-1,660035500012243860,027,01,073571,14:53:01,2010-05-
    21,01,true,APPROVED * =,1.00,M,112073-0_
    7,false,,,Ef3QO7bzCE3hTzmDqjvC5dMdl,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
mjr-res-indrefund-210510-251-1,660035500012243870,027,01,740332,14:53:08,2010-05-
    21,04,true,APPROVED * =,1.00,M,112074-0_
    7,false,,,dOeX9Hu7pRsLIJNcBphTTOmer,cc,,,bob@smith.com,this is my
    note,5454***5454,0812,1,,,,
```

Response Format for Vault Administrative Transactions

order_id, reserved, ResponseCode, [reserved], [reserved], TransTime, TransDate, [reserved], Complete, Message, [reserved],[reserved], [reserved], TimedOut, [reserved], [reserved], data_key, payment_type, cust_id, phone, email, note, pan, exp_date, crypt_type, avs_street_number, avs_street_name, avs_zip-code, [reserved], [reserved], sec, cust_first_name, cust_last_name, cust_address1, cust_address2, cust_city, cust_state, cust_zip, routing_num, account_num, check_num, account_type (Check Appendix C) [reserved], [reserved], [reserved], [reserved]

**EXAMPLE**

```
mjr-res-add-210510-251-1,null,001,null,null,14:52:49,2010-05-
    21,null,true,Successfully registered CC
    details.,null,null,null,false,,,we3u5tak7ce4UPiy1xRnDqqju,cc,moneris,phone,ema
    il,note,4242***4242,1111,7,,,,
mjr-res-delete-210510-251-1,null,001,null,null,14:53:13,2010-05-
    21,null,true,Successfully deleted CC
    details.,null,null,null,false,,,E08TCqdXsiOhDey9YZDa7QDMa,cc,moneris,phone,ema
    il,note,4242***4242,1111,7,,,,
```

### 5.2.3 VAU and ABU Response Information

The batch response file will be encrypted using your public GPG key. Once downloaded the merchant must decrypt it using their private GPG key. The response file will be in the “out” directory in your SFTP account. Moneris will add the extension ‘.out’ to the original file name, for example:

file.csv.abu.out

file.csv.vau.out

For a full description of each field please refer to Appendix - Response Fields. The .out file will contain a response for each request regardless if a match was found or not, please review the response file for all of the transactions.

**NOTE:** Separate VAU and ABU response files will be generated.

**Table 1: Response Fields - VAU and ABU Transactions**

| | |
|---|---|
| ABU Response | merchant_trans_id, old_pan, new_pan, old_exp_date, new_exp_date, mc_reason_id |
| VAU Response | merchant_trans_id, old_pan, new_pan, old_exp_date, new_exp_date, service_id, prev_sent |

#### 5.2.3.1 Time Frame for Receiving Response Files

To increase your chances of receiving the response file within the next day, you must submit your request files in the Moneris test environment or in the Moneris production environment as follows:

ABU:

- submit the request file between 1:00 AM and 3:30 PM EST
- the output file will be available for pick up by 9:00 AM EST the following day

VAU

- submit the request file between 1:00 AM and 7:00 AM EST
- the output file will be available for pick up by 9:00 AM EST the following day

### 5.2.4 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 .csv batch files to the test environment, first you must enter the correct credentials in your SFTP client as described in Configuring SFTP Client: What Do I Need to Do First? (page 59).

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 logins 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, usernames and passwords are different than the credentials you use in the production environment.

**Table 1: Test IDs**

| Store ID | Username | Password |
|---|---|---|
| store1 | DemoUser | password |
| store2 | DemoUser | password |
| store3 | 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 Testing Transactions with VAU and ABU

A testing environment is also available for you to connect to while you are testing uploading the GPG keys and the ".csv.abu" and/or ".csv.vau" request files.

For VAU and ABU testing you may use the following test card numbers with any future expiry date.

**Table 1: Test Card Numbers**

| Card Plan | Test Card Number |
|---|---|
| MasterCard | 5454545454545454 |

| Card Plan | Test Card Number |
|---|---|
| MasterCard | 5454545442424242 |
| Visa | 4242424242424242 |
| Visa | 4242424254545454 |

**NOTE:** We are unable to send test transactions onto the production ABU and VAU networks and thus responses are simulated.

To learn more on this subject, see also:

"Visa Account Updater and Automatic Billing Updater " on page 57

"Implementing VAU and ABU - Summary of Process" on page 8

1 Sending Transactions With VAU and ABU (page 1)

"VAU and ABU Response Information" on page 63

"GPG Keys" on page 9

# 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 8.

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 |
|---|---|---|
| order_id | 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.<br>Characters allowed: a-z A-Z 0-9 _ - : . @ spaces<br>**NOTE:** This variable also has field definition information exclusive to Vault. For more information, see the table Definitions of Required Fields - Vault Transactions. |
| pan | 20-character numeric | Credit Card Number with no spaces or dashes.<br>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. |
| exp_date | 4-character numeric | Expiry Date - format YYMM no spaces or slashes.<br>**NOTE:** This is reversed from the date format displayed on the physical card, MMYY. |
| track2 | string | This is a string that is retrieved |

| Variable Name | Size/Type | Description |
|---|---|---|
| | | from the mag swipe of a credit card by swiping the credit card through a card reader. It is part of a mag swipe/track2 transaction. |
| amount | 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point<br>**EXAMPLE:** 123456.78 | Amount of the transaction.<br>This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99 |
| crypt_type | 1-character alphanumeric | E-Commerce Indicator:<br>1 - Mail Order / Telephone Order - Single<br>2 - Mail Order / Telephone Order - Recurring<br>3 - Mail Order / Telephone Order - Instalment<br>4 - Mail Order / Telephone Order - Unknown Classification<br>5 - Authenticated E-commerce Transaction (VBV)<br>6 – Non Authenticated E-commerce Transaction (VBV)<br>7 - SSL enabled merchant<br>8 - Non Secure Transaction (Web or Email Based)<br>9 - SET non - Authenticated transaction |
| pos_code | 2-character numeric | Under normal presentment situations the value should be ‘00’.<br>If the solution is not “merchant and cardholder present” please call the support desk and we will provide the proper POS code. |
| txn_number | 255-character alphanumeric | Used when performing follow |

| Variable Name | Size/Type | Description |
|---|---|---|
| | | on transactions — this must be filled with the value that was returned as the txn_number in the response of the original transaction.<br>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. |
| cust_id (optional) | 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. |

# Appendix B Definitions of Response Fields

| Variable Name | Size/Type | Description |
|---|---|---|
| ReceiptId | 50-character alphanumeric | order_id specified in request |
| IssuerId | 50-character numeric | Credential on File response corresponding to the Issuer ID request variable |
| 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.<br>**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<br>< 50: Transaction approved<br>>= 50: Transaction declined<br>NULL: Transaction was not sent for authorization<br>If you would like further details on the response codes that are returned |

| Variable Name | Size/Type | Description |
|---|---|---|
| | | please see the Response Codes document available on the Moneris Developer Portal at https://developer.moneris.com |
| AuthCode | 8-character numeric | Authorization code returned from the issuing institution |
| TransTime | ##:##:## | Processing host time stamp (24 hour clock) |
| TransDate | yyyy-mm-dd | Processing host date stamp |
| TransType | 2-character numeric | Type of transaction that was performed<br>00 = Purchase, Purchase with Vault<br>01 = Pre-Authorization, Pre-Authorization with Vault<br>02 = Pre-Authorization Completion<br>04 = Refund, Independent Refund, Vault CC<br>11 = Purchase Correction |
| Complete | true/false | Transaction was sent to authorization host and a response was received |
| Message | 100-character alphanumeric | Response description returned from issuing institution. |
| TransAmount | | |
| CardType | 2-character alphanumeric | Card Type<br>M = MasterCard<br>V = Visa<br>AX = American Express<br>DC = Diners Card<br>NO = Novus / Discover<br>SE = Sears |
| TransID | 20-character alphanumeric | Gateway Transaction identifier |
| TimedOut | true/false | Transaction failed due to a process tim- |

| Variable Name | Size/Type | Description |
|---|---|---|
| | | ing out |
| Ticket | n/a | reserved |
| RecurSuccess | true/false | Indicates whether the transaction successfully registered |

## 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<br>< 50: Transaction approved<br>>= 50: Transaction declined<br>Vault Admin Responses:<br>001 Successfully registered CC details. Successfully deleted CC details.<br>983 Can not find previous<br>986 Incomplete: timed out<br>987 Invalid transaction<br>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.<br>Possible values:<br>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 – VAU and ABU

**Table 1: Definitions of Response Fields – VAU and ABU Transactions**

| Variable Name | Size/Type | Description |
|---|---|---|
| merchant_trans_id | 20-character alphanumeric | specified in request |
| old_pan | 20-character numeric | specified in request |

| Variable Name | Size/Type | Description |
|---|---|---|
|  | Variable |  |
| old_exp_date | 4-character numeric | specified in request |
| new_pan | 20-character numeric<br>variable | This field contains one of the following:<br>• The account number when new information is available<br>• All spaces for messages that indicate one of the following conditions:<br>• Closed Account<br>• Contact Cardholder<br>• Participating BIN, No Match<br>• Non-participating BIN, No Match<br>• 0 padded field |
| new_exp_date | 4-character numeric | This field contains one of the following:<br>• The expiration date when new information is available (YYMM format)<br>• Spaces for messages that indicate one of the following conditions:<br>• Closed Account<br>• Contact Cardholder<br>• Participating BIN, No Match<br>• Non-participating BIN, No Match<br>• 0 padded field |

## B.3  Definition of Response Fields – VAU Only

The following response fields apply to Visa Account Updater responses only.

**Table 1:  Definitions of Response Fields – VAU Only**

<table>
<tr><th>Variable Name</th><th>Size/Type</th><th>Description</th></tr>
<tr><td>service_id</td><td>1-character alphabetic</td><td>This response is returned by Visa<br>Possible values are:
<table>
<tr><th>Reason Code</th><th>Definition</th></tr>
<tr><td>A</td><td>Account number change message</td></tr>
<tr><td>C</td><td>Closed account advice</td></tr>
<tr><td>E</td><td>Expiration date change message</td></tr>
<tr><td>N</td><td>Non-participating BIN</td></tr>
<tr><td>Q</td><td>Contact cardholder advice</td></tr>
<tr><td>P</td><td>Participating BIN, no match</td></tr>
<tr><td>V</td><td>Match made, account number and expiration date unchanged</td></tr>
<tr><td>Z</td><td>No match found in the system</td></tr>
</table></td></tr>
<tr><td>prev_sent</td><td>Y/N</td><td>This field will return a value of “Y” which indicates that exactly the same account information was requested during the previous 40 days (the retention period), excluding the current day.</td></tr>
</table>

## B.4  Definitions of Response Fields - ABU Only

The following field applies to Automatic Billing Updater responses only.

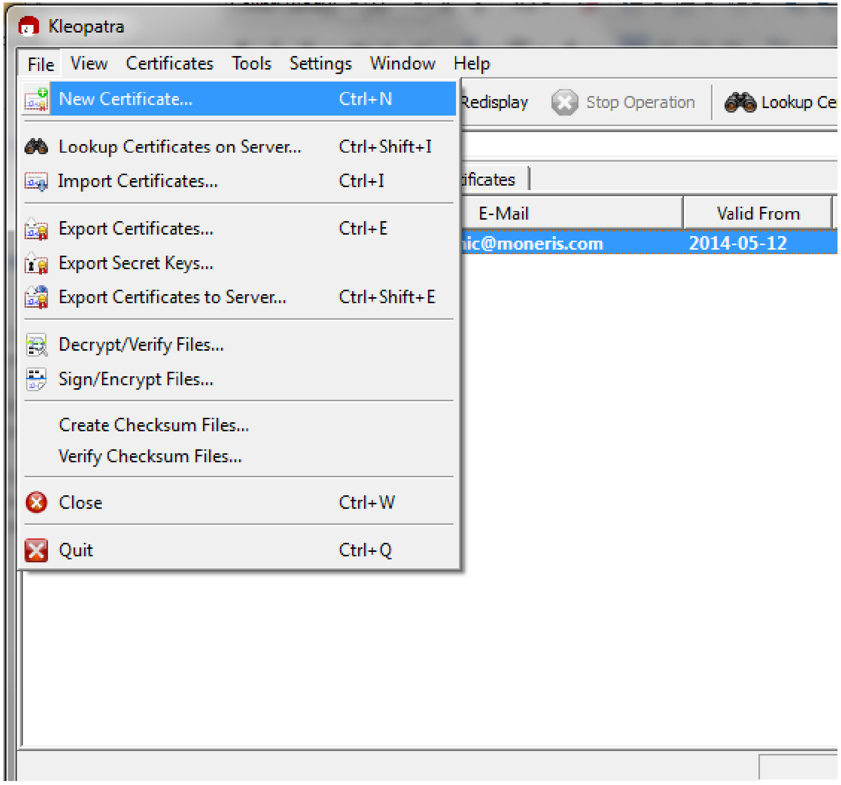
**Table 1:  Definitions of Response Fields – ABU transactions only**

| Variable Name | Size/Type | Description |
|---|---|---|
| mc_reason_id | 6-character alphanumeric | Response is returned by Master- |

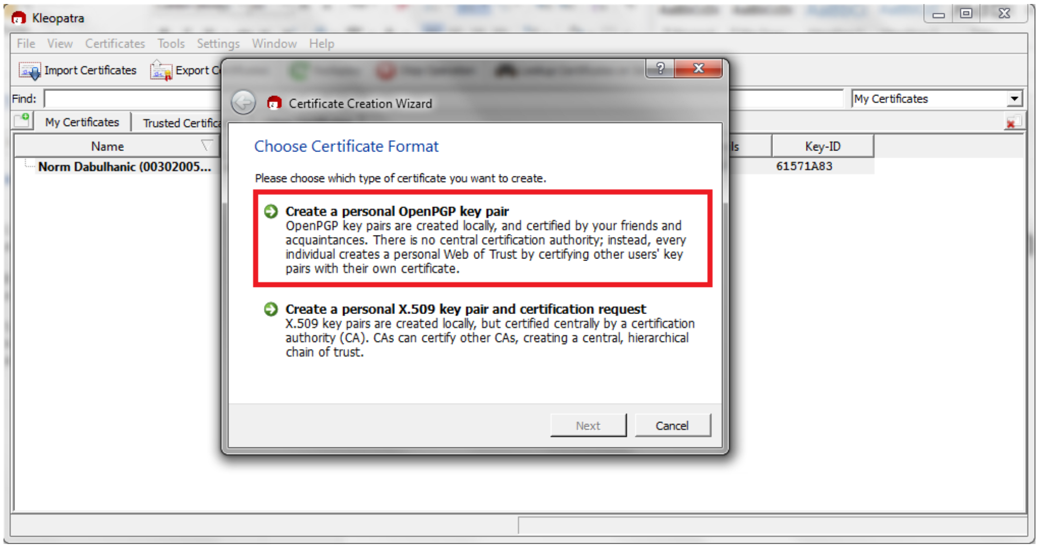
<table>
<thead>
<tr><th>Variable Name</th><th>Size/Type</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td></td><td></td><td>card<br>Possible values:
<table>
<thead>
<tr><th>Reason Code</th><th>Definition</th></tr>
</thead>
<tbody>
<tr><td>UPDATE</td><td>Update was made, updated data is returned</td></tr>
<tr><td>EXPIRY</td><td>Expiry date changed on the card</td></tr>
<tr><td>NOMATH</td><td>No match was found in Mastercard's system</td></tr>
<tr><td>000101</td><td>(ERROR) Non-numeric account number</td></tr>
<tr><td>000102</td><td>(ERROR) Bad check digit for account</td></tr>
<tr><td>000103</td><td>(ERROR) Invalid expiration date</td></tr>
<tr><td>000104</td><td>(ERROR) Merchant not registered</td></tr>
</tbody>
</table>
</td></tr>
</tbody>
</table>

# Appendix C  Generating a Key Using GPG4win

1. Install GPG4win and make sure GPA is also installed with the utility
2. Launch Kleopatra



3. Go to File menu – select New Certificate



Choose Certificate Format

Please choose which type of certificate you want to create.

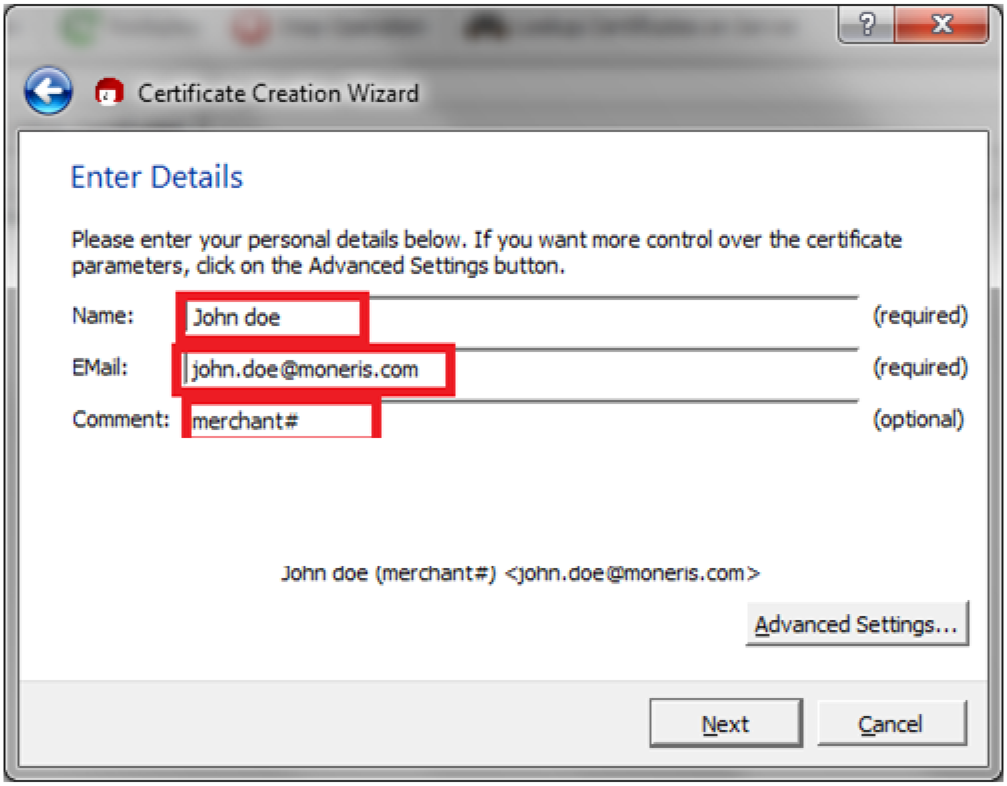
**Create a personal OpenPGP key pair**
OpenPGP key pairs are created locally, and certified by your friends and acquaintances. There is no central certification authority; instead, every individual creates a personal Web of Trust by certifying other users' key pairs with their own certificate.

**Create a personal X.509 key pair and certification request**
X.509 key pairs are created locally, but certified centrally by a certification authority (CA). CAs can certify other CAs, creating a central, hierarchical chain of trust.

4. Select Create a Personal OpenPGP key pair

5. Fill in the Full Name, Email, and Comment.

**NOTE:** The comment field is the most important and the 13 digit Moneris merchant number has to be placed in there.



Enter Details

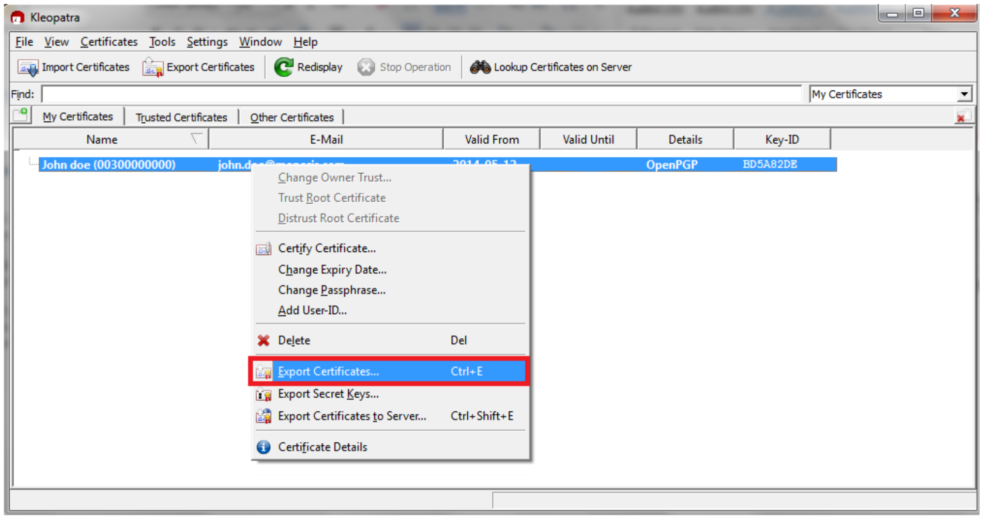
Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button.

| | | |
|---|---|---|
| Name: | John doe | (required) |
| EMail: | john.doe@moneris.com | (required) |
| Comment: | merchant# | (optional) |

John doe (merchant#) <john.doe@moneris.com>

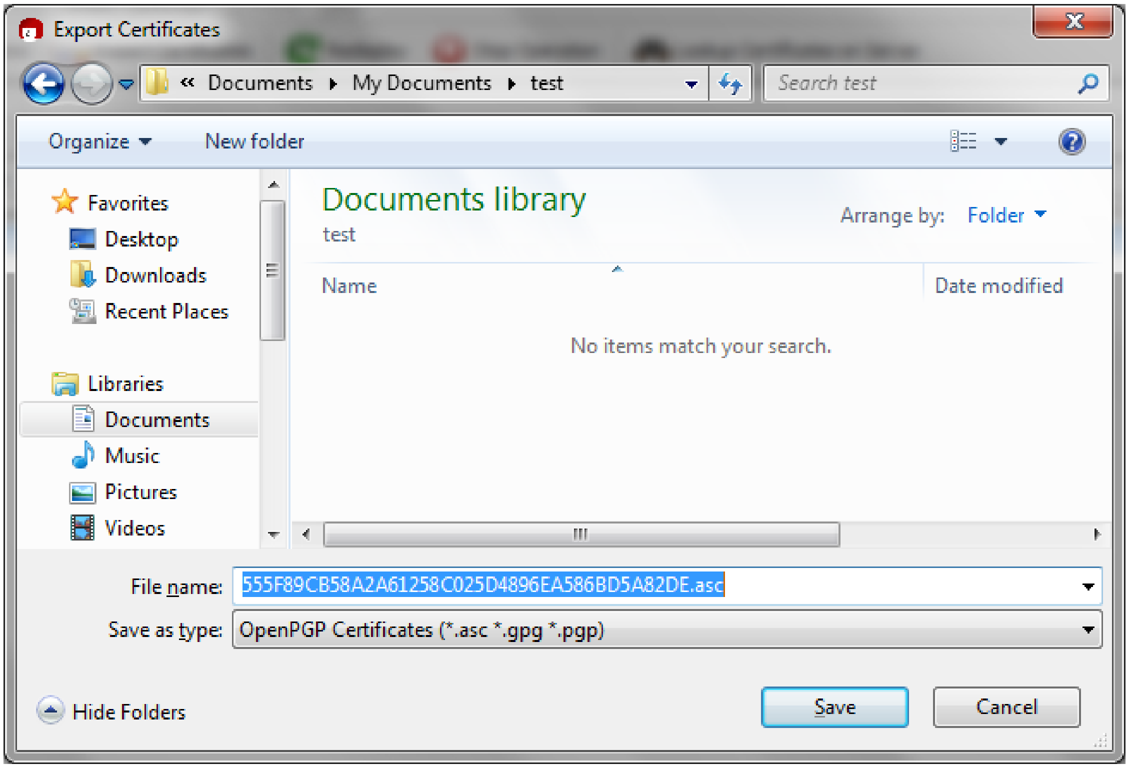
6. Click Create key

7. Enter the same passphrase twice
8. Click the finish button
9. In Kleopatra, right click the certificate you just created and select Export Certificate



Find:

| Name | E-Mail | Valid From | Valid Until | Details | Key-ID |
|---|---|---|---|---|---|
| John doe (00300000000) | john.d | | | OpenPGP | BD5A82DE |



File name: 555F89CB58A2A61258C025D4896EA586BD5A82DE.asc

Save as type: OpenPGP Certificates (*.asc *.gpg *.pgp)

10. Browse to the folder you are exporting the certificate to and Save the certificate

11. Open Windows Explorer and browse to the certificate you just created.

12. Rename the certificate to .gpg extension

13. Upload the file into the root directory of your account on our SFTP server.

# Appendix D  Error Messages

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

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

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

Message: 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