Moneris™

BE PAYMENT READY

Batch Upload Merchant Integration Guide

.csv

Canada only

Version 1.2.4 - October 2024

Table of Contents

Table of Contents **2**

Getting Help **4**

Changes in v1.2.4 ..... **5**

System and Skills Requirements **6**

1 Introduction to Batch Upload **7**
1.1 Implementing Your Batch Upload Solution 7
1.1.1 Implementing VAU and ABU - Summary of Process 7
1.1.1.1 GPG Keys 8

2 Transaction Types and Transaction Process Flows **10**
2.1 Basic Transactions 10
2.1.1 Basic Transactions Process Flow 11
2.1.2 Purchase 12
2.1.2.1 Purchase Transactions With Additional Features 14
2.1.2.2 Supplementary Purchase 14
2.1.2.3 Purchase with Credential on File 15
2.1.2.4 Supplementary Purchase with Credential on File 16
2.1.2.5 Purchase with Recurring Billing and Credential on File 17
2.1.3 Pre-Authorization 19
2.1.3.1 Pre-Authorization Transactions With Additional Features 21
2.1.3.2 Supplementary Pre-Authorization 21
2.1.3.3 Pre-Authorization with Credential on File 22
2.1.3.4 Supplementary Pre-Authorization with Credential on File 23
2.1.4 Pre-Authorization Completion 24
2.1.5 Force Post 27
2.1.5.1 Supplementary Force Post 29
2.1.6 Purchase Correction 30
2.1.7 Refund 31
2.1.8 Independent Refund 32
2.2 Mag Swipe Transactions 35
2.2.1 Sending Mag Swipe Transactions 35
2.2.1.1 Example – Batch Including Mag Swipe Transactions 36
2.2.2 Mag Swipe Transactions Process Flow 37
2.3 Vault Transactions 37
2.3.1 Vault Transactions Process Flow 38
2.3.2 Vault Administrative Transactions 39
2.3.2.1 Vault Add Credit Card 39
2.3.2.2 Vault Add Credit Card with Credential on File 41
2.3.2.3 Vault Update Credit Card 43
2.3.2.4 Vault Update Credit Card with Credential on File 46
2.3.2.5 Vault Delete 48
2.3.2.6 Example – Batch With Vault Admin Transactions 49
2.3.3 Vault Financial Transactions 50
2.3.3.1 Purchase with Vault 50
2.3.3.2 Purchase with Vault and Credential on File 52
2.3.3.3 Pre-Authorization with Vault 54
2.3.3.4 Pre-Authorization with Vault and Credential on File 56

2.3.3.5 Vault Independent Refund 59
2.3.3.6 Example – Batch With Vault Financial Transactions 61
2.4 Credential on File 62
2.4.1 About Credential on File Transactions – Batch Upload 62
2.4.2 Batch Upload Transactions with Credential on File 62
2.4.3 Deprecated Transactions Under Credential on File 63
2.4.4 Credential on File Info Object Request Fields 63
2.5 Recurring Billing 64
2.5.1 About Recurring Billing Transactions – Batch Upload 64
2.5.2 Sending Recurring Transactions 64
2.5.3 Recurring Billing Object and Request Fields 65
2.5.4 Example – Batch With Recurring Transactions 66
2.6 Visa Account Updater and Automatic Billing Updater 67

3 Sending Transactions and Receiving Responses 68
3.1 Creating a Batch File 68

4 Connecting to the Moneris Gateway to Upload Files 69
4.1 Configuring SFTP Client: What Do I Need to Do First? 69

5 Uploading a Batch File and Getting the Response 71
5.1 Filename Conventions for VAU and ABU 71
5.2 What Is Contained in a Transaction Response File? 71
5.2.1 Response Format for Basic Transactions 72
5.2.2 Response Formats for Vault Transactions 72
5.2.3 VAU and ABU Response Information 73
5.2.3.1 Time Frame for Receiving Response Files 74
5.2.4 Determining Whether a Transaction Was Successful 74

6 Testing Your Batch Upload Solution 76
6.1 Test Store Credentials 77
6.2 Testing Transactions with VAU and ABU 78
6.3 Test Card Numbers 79
6.4 Test Card Numbers for Level 2/3 79

7 Moving to Production 80
7.1 Activating a Store for Production 80

Appendix A Definition of Request Fields 81

Appendix B Definition of Response Fields 84
B.1 Definition of Response Fields – Vault 86
B.2 Definitions of Response Fields – VAU and ABU 87
B.3 Definition of Response Fields – VAU Only 88
B.4 Definitions of Response Fields - ABU Only 89

Appendix C Response Codes 91

Appendix D Generating a Key Using GPG4win 104

Appendix E Error Messages 108

Getting Help

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

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

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

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

Changes in v1.2.4

- Changed Moneris host addresses for testing and production in Configuring your SFTP Client topic
- Added Response Codes reference topic

Changes in v1.2.3

- added the request field **shipping indicator** to the Pre-Authorization Completion transaction (XML integrations only)

System and Skills Requirements

Before you start you will need to:

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

Additional requirements for VAU and ABU updating:

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

1 Introduction to Batch Upload

- 1.1 Implementing Your Batch Upload Solution
- 2 Transaction Types and Transaction Process Flows

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 about this,4.1 Configuring SFTP Client: What Do I Need to Do First?
3. **Uploading your batch transaction file and retrieving the response using your SFTP client**. To learn more about this, see 1 Uploading a Batch File and Receiving a Response.

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

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

To learn more about production, see 7 Moving to Production

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 Moneris customer service at:
>
> email: onlinepayments@moneris.com or call 1-866-319-7450
>
> You will be asked for your merchant name, merchant number and store ID.

1.1.1.1 GPG Keys

The VAU and ABU response files will be encrypted using the Gnu Privacy Guard (GPG) public key method. Below are the required steps. Please refer to Appendix D 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 request file will be encrypted. The response file will not be encrypted.

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

2 Transaction Types and Transaction Process Flows

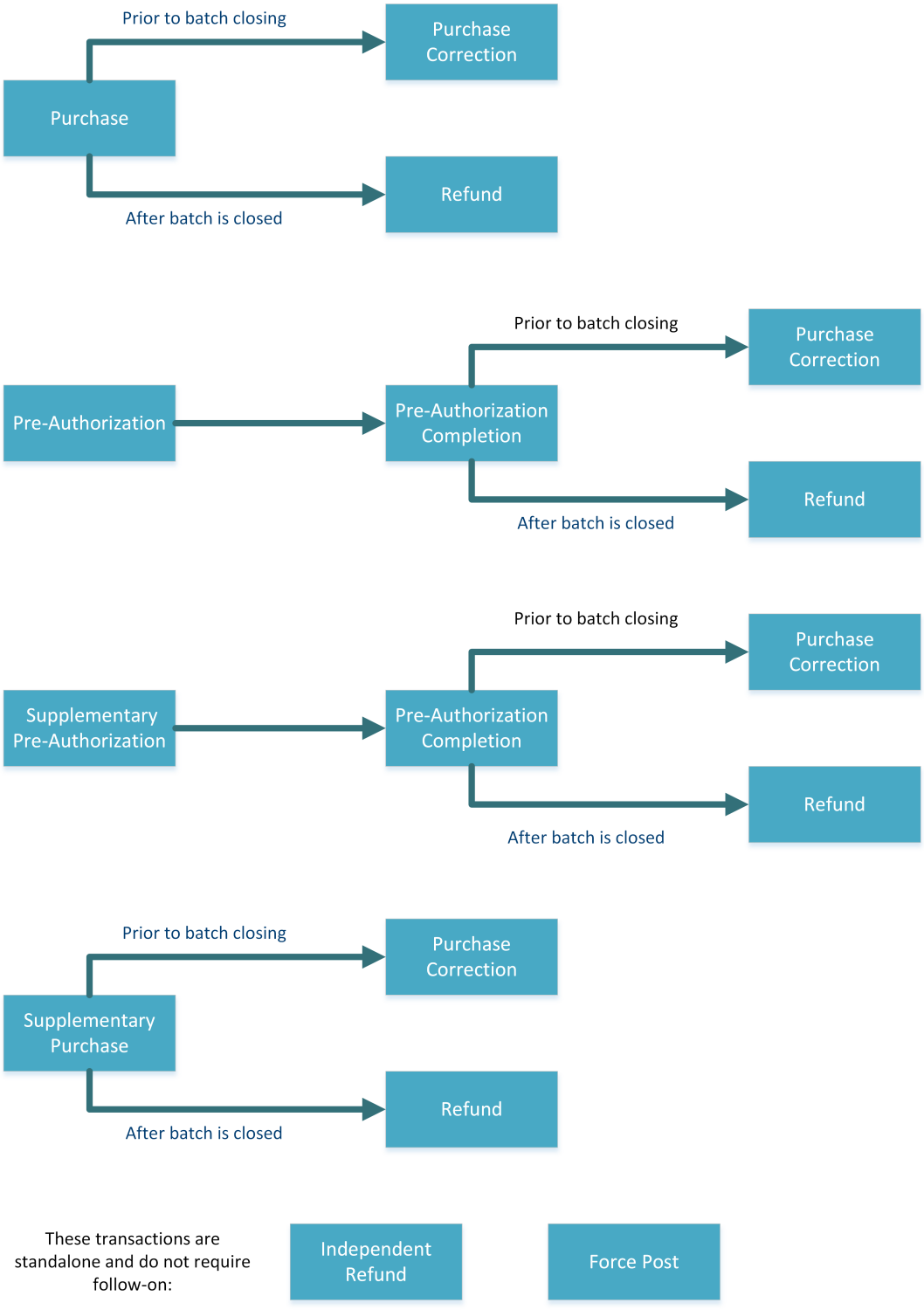
- 2.1 Basic Transactions
- 2.2 Mag Swipe Transactions
- 2.3 Vault Transactions
- 2.6 Visa Account Updater and Automatic Billing Updater

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

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

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 1 Purchase with Recurring Billing
Recurring Billing and Credential on File – See 2.1.2.5 Purchase with Recurring Billing and Credential on File
Credential on File – See 2.1.2.3 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 fields – Required Purchase

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

Variable Name	Type and Limits	Description
		must be the same as that of the original transaction.
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99 **NOTE:** For Purchase or Purchase with Vault requests that include Recurring Billing, the value of this field is the amount to bill immediately.
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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

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

Supplementary Purchase is a Purchase transaction with an optional Customer ID field included.

Required fields are the same as the Purchase transaction with the addition of the Customer ID field shown below. For descriptions of the other fields, see 2.1.2 Purchase.

CSV Request Format – Supplementary Purchase

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

CSV Response Format – Supplementary Purchase

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

Customer ID Request Field

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

Variable Name	Type and Limits	Description
		bership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center

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.

Required fields are the same as the basic Purchase transaction, with the addition of the Credential on File fields shown below. For descriptions of the other fields, see 2.1.2 Purchase.

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

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

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

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.

Required fields are the same as the Purchase transaction, with the addition of the Customer ID and Credential on File fields shown below.

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
```

Customer ID Request Field

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

Variable Name	Type and Limits	Description
Issuer ID issuer_id **NOTE:** This variable is required for all merchant-initiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character alphanumeric, variable	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator payment_indicator	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information payment_information	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible value: 2 - subsequent transactions (using previously stored payment details)

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.

Required fields are the same as the Purchase with Recurring Billing transaction, with the addition of the Credential on File fields shown below. For descriptions of the other fields, including the Recurring Billing fields, see 2.1.2 Purchase.

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

Variable Name	Type and Limits	Description
Issuer ID issuer_id **NOTE:** This variable is required for all merchant-initiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character alphanumeric, variable	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator payment_indicator	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information payment_information	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible value: 2 - subsequent transactions (using previously stored payment details)

2.1.3 Pre-Authorization

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

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 2.1.3.3 Pre-Authorization with Credential on File
Customer ID and Credential on File – See 2.1.3.4 Supplementary Pre-Authorization with Credential 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 fields – Required Pre-Authorization

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

Variable Name	Type and Limits	Description
	9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Trans-action (VBV) 7 - SSL enabled merchant

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

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

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

Required fields are the same as the Pre-Authorization transaction, with the addition of the Customer ID field. For descriptions of the other fields, see 2.1.3 Pre-Authorization.

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
```

Customer ID Request Field

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

Variable Name	Type and Limits	Description
		bership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center

2.1.3.3 Pre-Authorization with Credential on File

Pre-Authorization with Credential on File adds Credential on File fields to the Pre-Authorization transaction.

Required fields are the same as the Pre-Authorization transaction, with the addition of the Credential on File fields shown below.

CSV Request Format – Pre-Authorization with Credential on File

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

CSV Response Format – Pre-Authorization with Credential on File

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

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

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

2.1.3.4 Supplementary Pre-Authorization with Credential on File

Supplementary Pre-Authorization with Credential on File adds both the Customer ID and Credential on File information into the Pre-Authorization transaction request.

Required fields are the same as the Pre-Authorization transaction, with the addition of the Customer ID and Credential on File fields shown below. For descriptions of the other fields, see 2.1.3 Pre-Authorization.

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
```

Customer ID Request Field

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

Variable Name	Type and Limits	Description
		Can be searched from the Moneris Merchant Resource Center

Variable Name	Type and Limits	Description
Issuer ID issuer_id **NOTE:** This variable is required for all merchant-initiated transactions following the first one; upon sending the first transaction, the Issuer ID value is received in the transaction response and then used in subsequent transaction requests (Issuer ID does not apply for Discover or Union Pay).	*String* 15-character alphanumeric, variable	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator payment_indicator	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information payment_information	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible value: 2 - subsequent transactions (using previously stored payment details)

2.1.4 Pre-Authorization Completion

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

CSV Request Format – Pre-Authorization Completion

```
completion, order_id (from preauth), comp_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 fields – Required Pre-Authorization Completion

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

Variable Name	Type and Limits	Description
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

transaction request fields – Optional Pre-Authorization Completion

Variable Name	Type and Limits	Description
shipping indicator ship_indicator	*String* 1-character alphanumeric	Used to identify completion transactions that require multiple shipments, also referred to as multiple completions By default, if shipping indicator is not sent, the Pre-Authorization Completion is listed as final To indicate that the Pre-Authorization Completion is to be left open by the issuer as supplemental shipments or completions are pending, submit shipping indicator with a value of P Possible values: P – Partial F – Final

Variable Name	Type and Limits	Description

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 - See 2.1.5.1 Supplementary Force Post

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,
Complete, Message,amount, CardType, TxnNumber, TimedOut, reserved, reserved
```

transaction request fields – Required Force Post

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

Variable Name	Type and Limits	Description
	6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
Authorization code auth_code	*String* 8-character alphanumeric	Authorization code provided in the transaction response from the issuing bank
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

transaction request fields – Optional Force Post

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

2.1.5.1 Supplementary Force Post

Supplementary Force Post is the Force Post transaction with the Customer ID field included.

Required fields are the same as the Force Post transaction, with the addition of the Customer ID field shown below.

Optional Features

CSV Request Format – Supplementary Force Post

```
forcepost_supp, order_id, amount, pan, exp_date, auth_code, crypt_type, cust_id
```

CSV Response Format – Supplementary Force Post

```
order_id, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime, TransDate,Complete,Message,amount,CardType,TxnNumber,TimedOut,reserved,reserved
```

Customer ID Request Field

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

2.1.6 Purchase Correction

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

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

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 fields – Required Purchase Correction

Variable Name	Type and Limits	Description
Order ID order_id	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Transaction number txn_number	*String* 255-character alphanumeric	Used when performing follow on transactions — this must be filled with the value that was returned as the txn_number in the response of the original transaction. When performing a Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single

Variable Name	Type and Limits	Description
		2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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.

Values for the credit card number and expiry date are mandatory.

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 fields – Required Refund

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

Variable Name	Type and Limits	Description
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Transaction number txn_number	*String* 255-character alpha-numeric	Used when performing follow on transactions — this must be filled with the value that was returned as the txn_number in the response of the original transaction. When performing a Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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 fields – Required Independent Refund

Variable Name	Type and Limits	Description
Order ID order_id	*String* 50-character alphanumeric	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This

Variable Name	Type and Limits	Description
		field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

transaction request fields – Optional Independent Refund

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

2.2 Mag Swipe Transactions

Mag Swipe Purchase
The Mag Swipe Purchase transaction requires a credit card to be swiped to collect the track2 data. It then verifies funds on the customer's card, removes the funds and readies them for deposit into the merchant's account.

Mag Swipe Pre-Authorization
The Mag Swipe Pre-Authorization requires a credit card to be swiped to collect the track2 data. It then verifies and locks funds on the customer's credit card. The funds are locked for a specified amount of time, based on the card issuer. To retrieve the funds from a Mag Swipe Pre-Authorization so that they may be settled in the merchant's account a Mag Swipe Completion must be performed.

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

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

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

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

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

2.2.1 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

Transaction Type	Fields
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 track2_completion)
Mag Swipe Independent Refund (track2_ind_refund)	track2_ind_refund, order_id, cust_id, amount, track2, pan, exp_date, pos_code

2.2.1.1 Example – Batch Including Mag Swipe Transactions

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

> **NOTE:** In a Mag Swipe/track2 transaction, to preserve the order of the fields, you must still include a comma delimiter if you omit 'pan' or 'exp_date' fields.

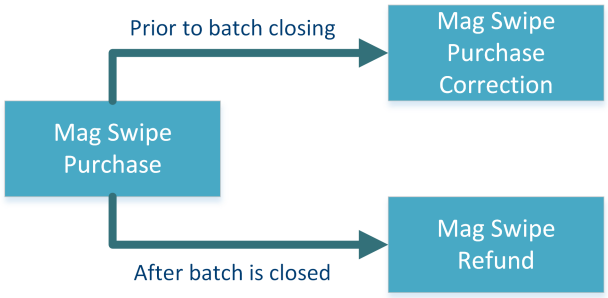
```
purchase,order_1_testing,13.00,4242424242424242,0304,1
purchase_supp,order_2_testing,2.00,4242424242424242,0908,1,customer_1
purchase,order_3_testing,13.00,5454545454545454,0403,1
preauth,order_4_testing,14.00,4242424242424242,0503,1
track2_purchase,track2_testing_1,cust_id,1.00,;5258984987184986=060610160910010606O2?,,,00
track2_preauth,track2_testing_2,cust_id,9.00,;5258984987184986=060610160910010606O2?,,,00
track2_ind_refund,track2_testing_3,cust_id,8.00,;5258984987184986=060610160910010606O2?,,,00
```

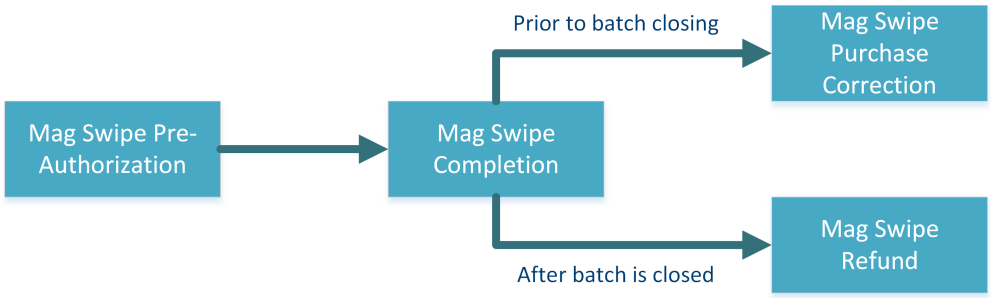
Corresponding Example Response for Mag Swipe Transaction

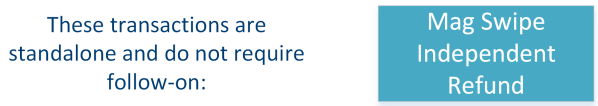
```
order_1_testing,660021810013368320,027,01,010460,11:04:27,2006-06-20,00,true,APPROVED *
=,13.00,V,63790-832-0,false,,null
order_2_testing,660021810013368330,027,01,009494,11:04:28,2006-06-20,00,true,APPROVED *
=,2.00,V,63791-833-0,false,,null
order_3_testing,660021810013368340,027,01,010461,11:04:30,2006-06-20,00,true,APPROVED *
=,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







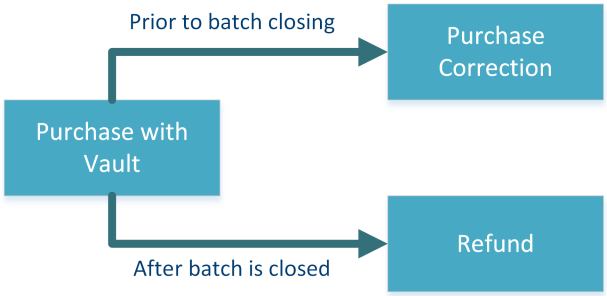
2.3 Vault Transactions

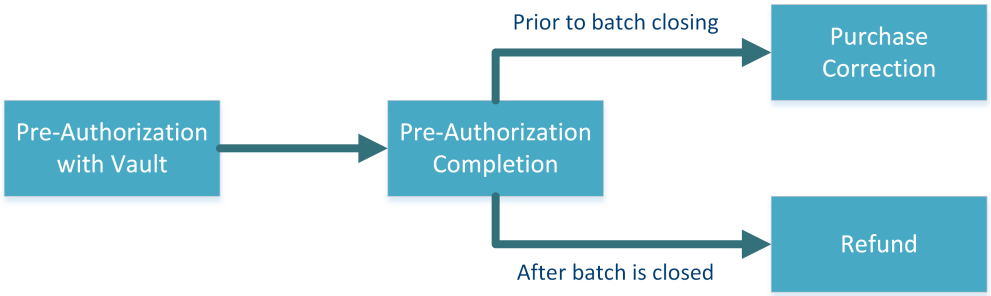
- 2.3.1 Vault Transactions Process Flow
- 2.3.2 Vault Administrative Transactions
- 2.3.3 Vault Financial Transactions

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





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

2.3.2 Vault Administrative Transactions

- 1 About Vault Administrative Transactions
- 2.3.2.1 Vault Add Credit Card
- 2.3.2.3 Vault Update Credit Card
- 2.3.2.5 Vault Delete

2.3.2.1 Vault Add Credit Card

This transaction has been deprecated for future use; going forward, use Vault Add Credit Card with Credential on File.

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
Credential on File – See 2.3.2.2 Vault Add Credit Card with Credential on File

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
```

Vault Add Credit Card transaction request fields – Required

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

Variable Name	Type and Limits	Description
		Correction transactions, the order ID must be the same as that of the original transaction.
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

Vault Add Credit Card transaction request fields – Optional

NOTE: Values for optional fields do not have to be sent in a transaction request, but the positional structure of CSV still requires the comma delimiters between each field regardless if a value is sent for the given field or not.

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

2.3.2.2 Vault Add Credit Card with Credential on File

Vault Add Credit Card with Credential on File is the Vault Add Credit Card transaction with the Credential on File fields included.

Required fields are the same as the Vault Add Credit Card transaction, with the addition of the Credential on File fields shown below. For descriptions of the other fields, see 2.3.2.1 Vault Add Credit Card.

CSV Request Format – Vault Add Credit Card with Credential on File

```
res_add_cc_cof, order_id, cust_id, phone, email, note, pan, exp_date, crypt_
type, issuer_id
```

CSV Response Format – Vault Add Credit Card with Credential on File

```
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, reserved
```

Vault Add Credit Card with Credential on File transaction request fields – Required

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

Variable Name	Type and Limits	Description
		Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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

Variable Name	Type and Limits	Description
NOTE: This variable is required for all merchant-initiated 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).	variable	Sent back in the response from the card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.

2.3.2.3 Vault Update Credit Card

Updates an existing Vault profile (referencing the profile's unique **data key**) with cardholder information.

Information contained within a credit card profile is updated as indicated by the submitted fields; if any field representing an item of cardholder information is not sent in this request, that item will remain unchanged in the profile.

If the Vault profile is being updated with a new credit card number, then you first need to send a Purchase, Pre-Authorization or Card Verification transaction, with the Credential on File Info object included, before performing Vault Update Credit Card. If the credit card number is not one of the profile items being updated, this step is not required.

Things to Consider:

- To update a specific element in the profile, set that element using the corresponding set method
- When updating a credit card number, first send a Purchase, Pre-Authorization, or Card Verification with the Credential on File Info object before sending this transaction; send the issuer ID received in the response in the subsequent Vault Update Credit Card request
- If the credit card number is not one of the profile items being updated, the Credential on File info object is not required

Optional Features

Customer ID – See Optional Fields table below
Credential on File – See 2.3.2.4 Vault Update Credit Card with Credential on File

CSV Request Format – Vault Update Credit Card

```
res_update_cc, order_id, data_key, 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
```

Vault Update Credit Card transaction request fields – Required

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

Vault Update Credit Card transaction request fields – Optional

NOTE: Values for optional fields do not have to be sent in a transaction request, but the positional structure of CSV still requires the comma delimiters between each field regardless if a value is sent for the given field or not.

Variable Name	Type and Limits	Description
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant
Customer ID cust_id	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Phone Number phone	*String* 30-character alphanumeric	Phone number of the customer
Note	*String*	Used for supplementary information

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

2.3.2.4 Vault Update Credit Card with Credential on File

> **NOTE:** Use this Credential on File transaction only when updating the credit card number (pan); for all other situations, use the basic Vault Update Credit Card transaction instead.

CSV Request Format – Vault Update Credit Card with Credential on File

```
res_update_cc_cof, order_id, data_key, cust_id, phone, email, note, pan, exp_
date, crypt_type, issuer_id
```

CSV Response Format – Vault Update Credit Card with Credential on File

```
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, reserved
```

Vault Update Credit Card with Credential on File transaction request fields – Required

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

Variable Name	Type and Limits	Description
		profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered

Vault Update Credit Card with COF transaction request fields – Optional

NOTE: Values for optional fields do not have to be sent in a transaction request, but the positional structure of CSV still requires the comma delimiters between each field regardless if a value is sent for the given field or not.

Variable Name	Type and Limits	Description
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Expiry Date exp_date	*String* 4-character numeric YYMM format	Expiry date with no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment

Variable Name	Type and Limits	Description
		4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant 8 - Non-secure Transaction (Web or Email Based) 9 - SET non-authenticated transaction
Customer ID cust_id	*String* 50-character alphanumeric	Merchant-defined value, used for additional identification purposes **EXAMPLE:** policy number, membership number, student ID, invoice number. Can be searched from the Moneris Merchant Resource Center
Phone Number phone	*String* 30-character alphanumeric	Phone number of the customer
Note note	*String* 30-character alphanumeric	Used for supplementary information
Email address email	*String* 30-character alphanumeric	Email address of the customer

2.3.2.5 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
```

Vault Delete transaction request fields – Required

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

2.3.2.6 Example – Batch With Vault Admin Transactions

```
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***
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,email,note,4242***
4242,1111,7,,,,
```

2.3.3 Vault Financial Transactions

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

2.3.3.1 Purchase with Vault

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

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

Optional Features

Customer ID – see Optional Fields table below
Credential on File – See 2.3.3.2 Purchase with Vault and 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
```

Purchase with Vault transaction request fields – Required

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

Variable Name	Type and Limits	Description
		Correction transactions, the order ID must be the same as that of the original transaction.
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Data key data_key	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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

2.3.3.2 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.

Required fields are the same as the Purchase with Vault transaction, with the addition of the Customer ID and Credential on File fields shown below. For descriptions of the other fields, see 2.3.3.1 Purchase with Vault.

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
```

Purchase with Vault and Credential on File

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

Variable Name	Type and Limits	Description
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Data key data_key	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

Variable Name	Type and Limits	Description
Issuer ID issuer_id	*String* 15-character alphanumeric, variable	Unique identifier for the cardholder's stored credentials Sent back in the response from the

Variable Name	Type and Limits	Description
NOTE: This variable is required for all merchant-initiated 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).		card brand when processing a transaction If the cardholder's credentials are being stored for the first time , you must use the Moneris Gateway API, Hosted Solutions, or the Merchant Resource Center to process the first transaction; only subsequent transactions can be carried out using Batch Upload.
Payment Indicator payment_indicator	*String* 1-character alphabetic	Indicates the intended or current use of the credentials Possible values for subsequent transactions: R - recurring U - unscheduled merchant-initiated transaction Z - unscheduled cardholder-initiated transaction
Payment Information payment_information	*String* 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible value: 2 - subsequent transactions (using previously stored payment details)

2.3.3.3 Pre-Authorization with Vault

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

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

Optional Features

Customer ID – See Optional Fields table below
Credential on File – See 2.3.3.4 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
```

Pre-Authorization with Vault transaction request fields – Required

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

Variable Name	Type and Limits	Description
		with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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

Customer Information Request Fields

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

2.3.3.4 Pre-Authorization with Vault and Credential on File

Pre-Authorization with Vault and Credential on File is a Pre-Authorization with Vault transaction with the Credential on File fields included.

Required fields are the same as the Pre-Authorization with Vault transaction, with the addition of the Credential on File fields shown below. For descriptions of the other fields, see 2.3.3.3 Pre-Authorization with Vault.

CSV Request Format – Pre-Authorization with Vault and Credential on File

```
res_preauth_cc_cof, order_id, cust_id, amount, data_key, crypt_type, issuer_
id, payment_indicator, payment_information
```

CSV Response Format – Pre-Authorization 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
```

Pre-Authorization with Vault and Credential on File transaction request fields – Required

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

Variable Name	Type and Limits	Description
		The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

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

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

2.3.3.5 Vault 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 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

Vault Independent Refund transaction request fields – Required

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

Variable Name	Type and Limits	Description
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Data key data_key	*String* 25-character alphanumeric	The data key is the token that points to a previously stored profile; a profile identifier that all future financial Vault transactions (i.e., they occur after the profile was registered by a Vault Add Credit Card or Vault Tokenize Credit Card transaction) will use to associate with the saved information The data key is generated by Moneris, and is returned to the merchant (via the Receipt object) when the profile is first registered
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order - Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant

Vault Independent Refund transaction request fields – Optional

> **NOTE:** Values for optional fields do not have to be sent in a transaction request, but the positional structure of CSV still requires the comma delimiters between each field regardless if a value is sent for the given field or not.

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

2.3.3.6 Example – Batch With Vault Financial Transactions

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

```
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
- 2.4.4 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

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

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

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

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

2.4.2 Batch Upload Transactions with Credential on File

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

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

2.4.3 Deprecated Transactions Under Credential on File

The following transactions are deprecated because of the Credential on File rules. Going forward, use the transaction types that are denoted by "Credential on File" instead.

Purchase
Pre-Authorization
Purchase with Recurring Billing
Supplementary Purchase
Supplementary Pre-Authorization
Vault Add Credit Card

2.4.4 Credential on File Info Object Request Fields

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

Variable Name	Type and Limits	Description
		Possible value: 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
- 2.5.4 Example – Batch With 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 Recurring Billing Object and Request Fields

Recurring Billing Transactions Request Fields

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

Variable Name	Type and Limits	Description
recur_unit	*String* day, week, month, eom	The unit that you wish to use as a basis for the Interval. This can be set as day, week or month. Then using the "period" field you can configure how many days, weeks, months between billing cycles.
period	*String* 0 – 999 characters, numeric	This is the number of recur_ units you wish to pass between billing cycles. Example : period = 3, recur_unit=month -> Card will be billed every 3 months. period = 4, recur_unit=weeks -> Card will be billed every 4 weeks. period = 45, recur_unit=day -> Card will be billed every 45 days. Please note that the total duration of the recurring billing transaction should not exceed 5-10 years in the future.
start_date	*String* YYYY/MM/DD	This is the date on which the first charge will be billed. The value must be in the future. It cannot be the day on which the transaction is being sent. If the transaction is to be billed immediately the start_now feature must be set to true and the start_date should be set at the desired interval after today.

Variable Name	Type and Limits	Description
start_now	*String* true / false	When a charge is to be made against the card immediately start_now should be set to 'true'. If the billing is to start in the future then this value is to be set to 'false'. When start_now is set to 'true' the amount to be billed immediately may differ from the recur amount billed on a regular basis thereafter.
recur_amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the recurring transaction. This must contain 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99. This is the amount that will be billed on the start_date and every interval thereafter.
num_recurs	*String* 1 – 99 characters, numeric	The number of times to recur the transaction.

2.5.4 Example – Batch With Recurring Transactions

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

```
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 credential 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:

- Brand conversion
- Expiry date changes
- Card Upgrades
- Re-issued cards due to fraud or new programs
- 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 Sending Transaction Requests

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 69.

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

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

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	gwutilstest.moneris.com	gwutils.moneris.com
Port	22	22
Server Type	SFTP using SSH2	SFTP using SSH2
Username	store3	Provided by Moneris (please refer to NOTE)
Password	store3	Provided by Moneris (please refer to NOTE)

NOTE:
To get your test and/or production SFTP username and password, please contact Moneris customer service at:

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

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

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

To learn more on this subject, see also:

5 Uploading a Batch File and Getting the Response

7 Moving to Production

5 Uploading a Batch File and Getting the Response

- 5.1 Filename Conventions for VAU and ABU
- 5.2 What Is Contained in a Transaction Response File?

When uploading a file it must be put in your default root directory. Do not put it in the processing directory, as it will not get processed.

When a file is complete the response file will be put in the /out folder. The filename will be identical to what was uploaded but will be appended with the extension ".out", as shown in the following table.

Table 1 Example of Request and Response Filename Usage

Request File	Response File
testing_june_20.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?

- "Response Format for Basic Transactions" on the next page
- "Response Formats for Vault Transactions" on the next page

- "VAU and ABU Response Information" on the facing page
- "Determining Whether a Transaction Was Successful" on page 74

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 "Definition of Response Fields" on page 84.

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,phon
    e,email,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,phon
    e,email,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

- 6.2 Testing Transactions with VAU and ABU

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

To verify that these transactions have been properly received and processed, please refer to the .out file as well as logging into the Merchant Resource Center to view reports of your transactions.

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

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

> **NOTE:** These test store IDs, 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 Test Store Credentials

For testing purposes, you can either use the pre-existing test stores with the corresponding test API tokens, or you can create your own unique test API token and a unique test store where you will only see your own transactions. If you want to use pre-existing stores, use the test credentials provided in the following tables.

Table 1: Test Server Credentials - Canada

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

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

6.2 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
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 67

"Implementing VAU and ABU - Summary of Process" on page 7

1 Sending Transactions With VAU and ABU (page 1)

"VAU and ABU Response Information" on page 73

"GPG Keys" on page 8

6.3 Test Card Numbers

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

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

Card Plan	Test Card Number
Mastercard	5454545454545454
Visa	4242424242424242
Amex	373599005095005
JCB	3566007770015365
Diners	36462462742008
Track2	5258968987035454=06061015454001060101?
Discover	6011000992927602
UnionPay	6250944000000771

6.4 Test Card Numbers for Level 2/3

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

Card Brand	Test Card Number
Mastercard	5454545442424242
Visa	4242424254545454
Amex	373269005095005

7 Moving to Production

- 1 How Do I Activate My Store?

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

The process of sending transaction requests and receiving the responses is nearly identical in production as it is in testing. To review the primary steps in the batch upload process again, see "Implementing Your Batch Upload Solution" on page 7.

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

> **NOTE:**
> To receive your Production SFTP Username and Password, please call Moneris at 1-866-319-7450

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 Definition of Request Fields

Table 1 Definition of Request Fields – Basic and Mag Swipe Transactions

Variable Name	Type and Limits	Description
Amount amount	*String* 9-character decimal. Up to 6-character numeric + 2-character numeric after the decimal point **EXAMPLE:** 123456.78	Amount of the transaction. This must contain at least 3 digits with two penny values. The minimum value passed can be 0.01 and the maximum 999999.99
Credit card number pan	*String* 20-character numeric	Credit Card Number with no spaces or dashes. Most credit card numbers today are 16 digits in length but some 13 digits are still accepted by some issuers. This field has been intentionally expanded to 20 digits in consideration of future expansion and/or potential support of private label card ranges.
Customer ID cust_id (optional)	*String* 50-character alphanumeric	This is an optional field that can be sent as part of a Purchase or Pre-Authorization request. It is searchable from the Merchant Resource Center. It is commonly used for policy number, membership number, student ID or invoice number.
E-commerce indicator crypt_type	*String* 1-character alphanumeric	E-commerce Indicator possible values: 1 - Mail Order / Telephone Order - Single 2 - Mail Order / Telephone Order - Recurring 3 - Mail Order / Telephone Order -

Variable Name	Type and Limits	Description
		Instalment 4 - Mail Order / Telephone Order - Unknown Classification 5 - Authenticated E-commerce Transaction (VBV) 6 – Non Authenticated E-commerce Transaction (VBV) 7 - SSL enabled merchant
Expiry date exp_date	*String* 4-character numeric	Expiry Date - format YYMM no spaces or slashes. **NOTE:** This is reversed from the date format displayed on the physical card, MMYY.
Order ID order_id	*String* 50-character alphanumeric	Merchant defined unique transaction identifier — must be unique for every purchase, preauth and ind_refund attempt. For refund,completion and purchasecorrection, the order_id must reference the original transaction. Characters allowed: a-z A-Z 0-9 _ - : . @ spaces **NOTE:** This variable also has field definition information exclusive to Vault. For more information, see the table Definitions of Required Fields - Vault Transactions.
POS Code pos_code	*String* 2-character numeric	Under normal presentment situations the value should be '00'. If the solution is not "merchant and cardholder present" please call the support desk and we will provide the proper POS code.

Variable Name	Type and Limits	Description
shipping indicator ship_indicator	*String* 1-character alphanumeric	Used to identify completion transactions that require multiple shipments, also referred to as multiple completions By default, if shipping indicator is not sent, the Pre-Authorization Completion is listed as final To indicate that the Pre-Authorization Completion is to be left open by the issuer as supplemental shipments or completions are pending, submit shipping indicator with a value of P Possible values: P – Partial F – Final
Track 2 track2	*String*	This is a string that is retrieved from the mag swipe of a credit card by swiping the credit card through a card reader. It is part of a mag swipe/track2 transaction.
Transaction number txn_number	*String* 255-character alphanumeric	Used when performing follow on transactions — this must be filled with the value that was returned as the txn_number in the response of the original transaction. When performing a Pre-Authorization Completion this must reference the Pre-Authorization. When performing a Refund or a Purchase Correction this must reference the Pre-Authorization Completion or the Purchase.

Appendix B Definition of Response Fields

Variable Name	Size/Type	Description
AuthCode	8-character numeric	Authorization code returned from the issuing institution
CardType	2-character alphanumeric	Card Type M = MasterCard V = Visa AX = American Express DC = Diners Card NO = Novus / Discover SE = Sears
Complete	true/false	Transaction was sent to authorization host and a response was received
IssuerId	15-character numeric	Credential on File response corresponding to the Issuer ID request variable
Message	100-character alphanumeric	Response description returned from issuing institution.
ReceiptId	50-character alphanumeric	order_id specified in request
RecurSuccess	true/false	Indicates whether the transaction successfully registered
ReferenceNum	18-character numeric	The reference number is an 18 character string that references the terminal used to process the transaction as well as the shift, batch and sequence number. This data is typically used to reference transactions on the host systems and must be displayed on any receipt presented to the customer. This information should be stored by the merchant. **EXAMPLE:** The following

Variable Name	Size/Type	Description
		illustrates the breakdown of this field where "640123450010690030" is the reference number returned in the message, "64012345" is the terminal id, "001" is the shift number, "069" is the batch number and "003" is the transaction number within the batch. Moneris Host Transaction identifier.
ResponseCode	3-character numeric	Transaction Response Code < 50: Transaction approved >= 50: Transaction declined NULL: Transaction was not sent for authorization If you would like further details on the response codes that are returned please see the Response Codes document available on the Moneris Developer Portal at https://developer-.moneris.com
Ticket	n/a	reserved
TimedOut	true/false	Transaction failed due to a process timing out
TransAmount		
TransDate	yyyy-mm-dd	Processing host date stamp
TransID	20-character alphanumeric	Gateway Transaction identifier
TransTime	hh:mm:ss (24 hour clock)	Processing host time stamp
TransType	2-character numeric	Type of transaction that was performed 00 = Purchase, Purchase with Vault 01 = Pre-Authorization, Pre-Authorization with Vault

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

B.1 Definition of Response Fields – Vault

Variable Name	Type and Limits	Description
ReceiptId	50-character alphanumeric	order_id specified in request
[reserved]	n/a	Future use
ResponseCode	3-character numeric	Transaction Response Code < 50: Transaction approved >= 50: Transaction declined Vault Admin Responses: 001 Successfully registered CC details. Successfully deleted CC details. 983 Can not find previous 986 Incomplete: timed out 987 Invalid transaction Null Error: Malformed XML
TransTime	hh:mm:ss (24 hour clock)	Processing host time stamp
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
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

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

B.2 Definitions of Response Fields – 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 Variable	specified in request
old_exp_date	4-character numeric	specified in request

Variable Name	Size/Type	Description

new_pan	20-character numeric
	
	variable
		This field contains one of the following:
		
		• The account number when new information is available
		• All spaces for messages that indicate one of the following conditions:
		  • Closed Account
		  • Contact Cardholder
		  • Participating BIN, No Match
		  • Non-participating BIN, No Match
		
		• 0 padded field

new_exp_date	4-character numeric	This field contains one of the following:
		
		• The expiration date when new information is available (YYMM format)
		• Spaces for messages that indicate one of the following conditions:
		  • Closed Account
		  • Contact Cardholder
		  • Participating BIN, No Match
		  • Non-participating BIN, No Match
		
		• 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

Variable Name	Size/Type	Description
service_id	1-character alphabetic	This response is returned by Visa Possible values are: **Reason Code** / **Definition** A – Account number change message C – Closed account advice E – Expiration date change message N – Non-participating BIN Q – Contact cardholder advice P – Participating BIN, no match V – Match made, account number and expiration date unchanged Z – No match found in the system
prev_sent	Y/N	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.

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 Mastercard Possible values:

Variable Name	Size/Type	Description
		Reason Code / **Definition** UPDATE — Update was made, updated data is returned EXPIRY — Expiry date changed on the card NOMATH — No match was found in Mastercard's system 000101 — (ERROR) Non-numeric account number 000102 — (ERROR) Bad check digit for account 000103 — (ERROR) Invalid expiration date 000104 — (ERROR) Merchant not registered

Appendix C Response Codes

Approved Response Codes

Response Code	Messages
000	Approved, Account Balances Included (Balance Inquiry), No Reason to Decline Approved (Balances) File Processed/Successful transaction with fault
001	Approved, Account Balances Not Included Approved – No Balances/Approved or completed successfully VIP Approved (No Balances)/Advice Acknowledged – Financial Liability Accepted
002	Approved, country club
003	Approved, maybe more ID
004	Approved, pending ID (sign paper draft)
005	Approved, blind
006	Approved, VIP
007	Approved, administrative transaction
008	Approved, national NEG file hit OK
009	Approved, commercial
010	Approved for partial amount
023	Amex - credit approval
024	Amex 77 - credit approval
027	Transaction already reversed

Response Code	Messages
028	VIP Credit Approved
029	Credit Response Acknowledgement
900	Global Error
901	Invalid URL
902	Malformed XML

Declined Response Codes

Response Code	Messages
050	Do Not Honor Decline Refer to card issuer ID certification fails Deny – Do not Honour Card not initialized Declined: Deny – Unacceptable Fee Unable to locate original transaction Suspected Fraud Deny – Card Acceptor Call Acquirer’s Security Dep Amount Not Reconciled – Totals Provided ATM/POS terminal number cannot be located MAC failed Declined: MAC failed Reserved Security processing failure No arrears (transaction receipt not printed)

Response Code	Messages
	Invalid File Type No such File File Locked Unsuccessful Incorrect File Length File Decompression Error File Name Error File cannot be received Deny – Do Not Honour
051	Expired Card
052	PIN retries exceeded PIN try limit exceeded Allowable number of PIN tries exceeded
053	No sharing
054	No security module
055	Invalid transaction
056	No Support/Transaction Not Permitted to Acquirer Tran Not Supported by FI/Not Supported by Receiver
057	Lost or stolen card
058	Invalid status
059	Deny (Keep Card) – Restricted Card Restricted Card
060	No Chequing account No Savings Account
061	No PBF

Response Code	Messages
062	PBF update error
063	Invalid authorization type
064	Bad Track 2
065	Adjustment not allowed
066	Invalid credit card advance increment
067	Invalid transaction date
068	PTLF error
069	Bad Message Error/No CVM Results Bad message – edit error/Format error
070	No IDF Invalid Issuer Invalid Issuer/Deny – Issuer/Bank Not Found
071	Invalid route authorization Unable to route/Financial institution or intermediate network facility cannot be found for routing Invalid Rout to Auth /Incorrect IIN
072	Card on National NEG file
073	Invalid route service (destination)
074	Unable to authorize Re-enter Transaction Transaction Cannot be Completed Deny – Security Violation Deny – Violation of Law System problem - ask cardholder to insert card in chip card reader Merchant Link not logged on (Network Management Logon required)
075	Invalid PAN length

Response Code	Messages
076	Low funds
077	Pre-auth full
078	Duplicate transaction Duplicate transaction/Request in progress
079	Maximum online refund reached
080	Maximum offline refund reached
081	Maximum credit per refund reached
082	Number of times used exceeded
083	Maximum refund credit reached
084	Duplicate transaction - authorization number has already been corrected by host
085	Inquiry not allowed
086	Over floor limit
087	Maximum number of refund credit by retailer
088	Place call
089	CAF status inactive or closed
090	Referral file full
091	NEG file problem
092	Advance less than minimum
093	Delinquent
094	Over table limit
095	Amount over maximum Amt Over Max/Transaction amount limit exceeded
096	PIN required

Response Code	Messages
097	Mod 10 check failure
098	Force Post
099	Bad PBF

Referral Response Codes

Response Code	Messages
100	Unable to process transaction Invalid Request. Contact Moneris Client POS Certification for repeat declines. Network Unavailable System Malfunction
101	Place call
102	Refer – Call Expired Card Card Acceptor Contact Call Card Accpt Acq Secur
103	NEG file problem
104	CAF problem
105	Card not supported
106	Amount over maximum
107	Over daily limit
108	CAF Problem
109	Advance less than minimum
110	Number of times used exceeded
111	Delinquent

Response Code	Messages
112	Over table limit
113	Timeout
115	PTLF error
121	Administration file problem
122	Unable to validate PIN: security module down

System Error Response Codes

Response Code	Messages
150	Invalid Service Code/Merchant Merchant Not On File Merchant Not on File/Invalid Merchant
200	Invalid account Invalid Card Number Invalid Account/Deny – No Account Type Requested
201	Incorrect PIN Invalid PIN/Incorrect personal identification number PIN Block Error
202	Advance less than minimum
203	Administrative card needed
204	Amount over maximum
205	Invalid Advance Amount Original Amnt Incorrect Bad message/Invalid Amount Original transaction amount error
206	CAF not found

Response Code	Messages
	Invalid “to” account Invalid “from” account Invalid account
207	Invalid transaction date
208	Invalid expiration date
209	Invalid transaction code
210	PIN key sync error
212	Destination not available
251	Error on cash amount
252	Debit not supported

American Express Response Codes (Declines)

Response Code	Messages
426	AMEX - Denial 12
427	AMEX - Invalid merchant
429	AMEX - Account error
430	AMEX - Expired card
431	AMEX - Call Amex
434	AMEX - Call 03 Note: Invalid CVD (CID)
435	AMEX - System down
436	AMEX - Call 05
437	AMEX - Declined
438	AMEX - Declined

Response Code	Messages
439	AMEX - Service error
440	AMEX - Call Amex
441	AMEX - Amount error

Credit Card Response Codes (Declines)

Response Code	Messages
408	CREDIT CARD - Card use limited - Refer to branch
475	CREDIT CARD - Invalid expiration date
476	CREDIT CARD - Invalid transaction, rejected No Credit Account Invalid transaction/Invalid related transactions Unable to process/Suspected malfunction; related transaction error Unable to Authorize: Cut off is in process Issuer not capable to process Switch system malfunction Issuer response not received by CUPS Unable to Authorize/Illegal Status of Acquirer
477	CREDIT CARD - Refer Call/Invalid Card Number Invalid card number (no such account) Deny – Card Not Found Items not on Bankbook beyond limit, declined/Invalid card number
478	CREDIT CARD - Decline, Pick up card, Call
479	CREDIT CARD - Decline, Pick up card
480	CREDIT CARD - Decline, Pick up card
481	CREDIT CARD - Decline

Response Code	Messages
	Transaction not allowed to be processed by cardholder Low funds/Insufficient Balance Invalid Transaction Transaction not allowed to be processed by merchant
482	CREDIT CARD - Expired Card
483	CREDIT CARD – Refer/Refer to Issuer Deny – Card Acceptor Contact Acquirer
484	CREDIT CARD - Expired card - refer
485	CREDIT CARD - Not authorized
486	CREDIT CARD - CVV Cryptographic error
487	CREDIT CARD - Invalid CVV
489	CREDIT CARD - Invalid CVV
490	CREDIT CARD - Invalid CVV
492	System problem - ask cardholder to insert card in chip card reader Withdrawal count exceeded

System Decline Response Codes

Response Code	Messages
800	Bad format
801	Bad data
802	Invalid Clerk ID
809	Bad close
810	System timeout
811	System error
821	Bad response length

Response Code	Messages
877	Invalid PIN block
878	PIN length error
880	Final packet of a multi-packet transaction
881	Intermediate packet of a multi-packet transaction
889	MAC key sync error
898	Bad MAC value
899	Bad sequence number - resend transaction
900	Capture - PIN Tries Exceeded
901	Capture - Expired Card
902	Capture - NEG Capture
903	Capture - CAF Status 3
904	Capture - Advance < Minimum
905	Capture - Num Times Used
906	Capture - Delinquent
907	Capture - Over Limit Table
908	Capture - Amount Over Maximum Capture - Capture Pick up Card Suspected Fraud Hard Capture Deny – Keep Card: Special Conditions Expired Card Fraud Card Acceptor Call Acquirer’s

Response Code	Messages
	Do Not Honour
950	Admin card is not enabled on Merchant profile

Velocity Reject Response Code (Non-RBC Response)

Response Code	Message
959	Decline

Admin Response Codes

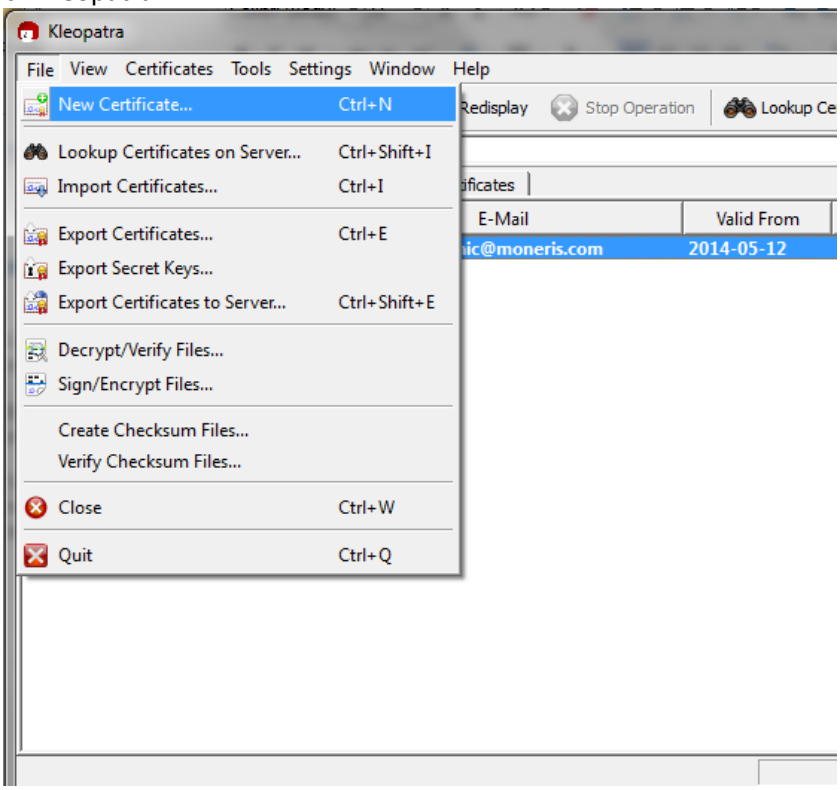
Response Code	Messages
960	Initialization Failure - No Match on Merchant ID
961	Initialization Failure - No Match on PED ID
962	Initialization Failure - No match on Printer ID
963	No match on Poll code
964	Initialization Failure - No match on Concentrator ID
965	Invalid software version number
966	Duplicate terminal name
970	Terminal/Clerk table full
983	Clerk Totals Unavailable: selected Clerk IDs do not exist or have zero totals
989	MAC Error on Transaction 95 (Initialization and Handshake), most often, this indicates that the wrong keys have been injected into a device/KMAC Sync Error

EMV Reversal Request Codes

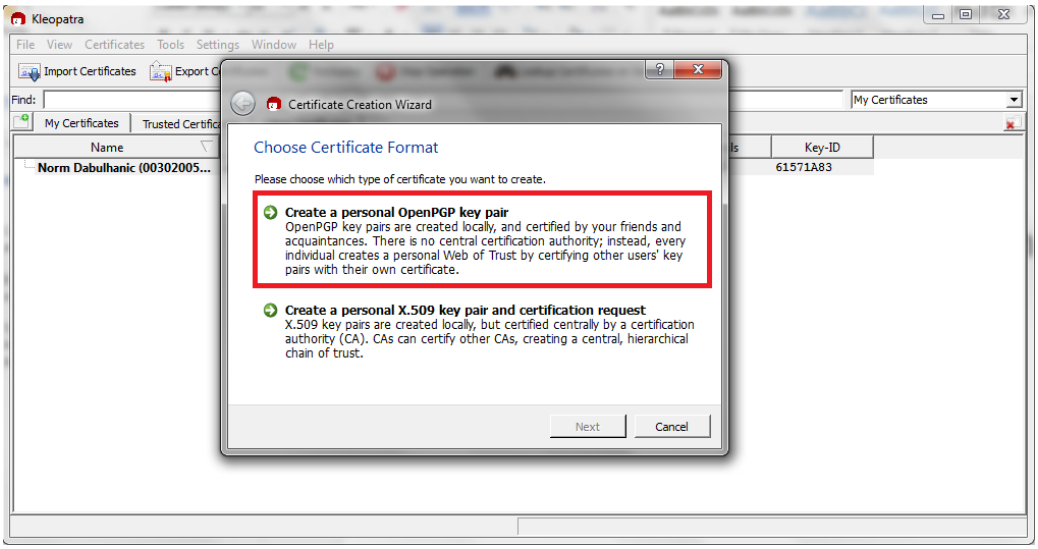
Response Code	Messages
990	Chip card declines a host approved transaction

Response Code	Messages
991	Chip card removed before ICC communications are completed

Appendix D Generating a Key Using GPG4win

1. Install GPG4win and make sure GPA is also installed with the utility
2. Launch Kleopatra
3. 

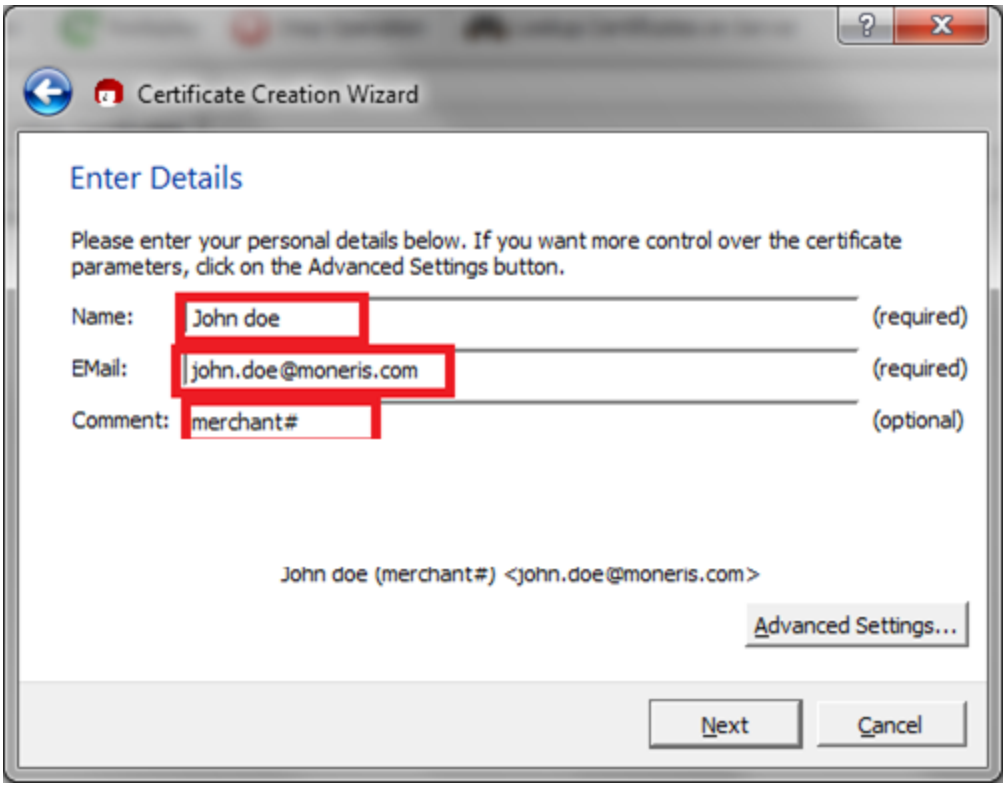
   Go to File menu – select New Certificate

4. 

   Select Create a Personal OpenPGP key pair

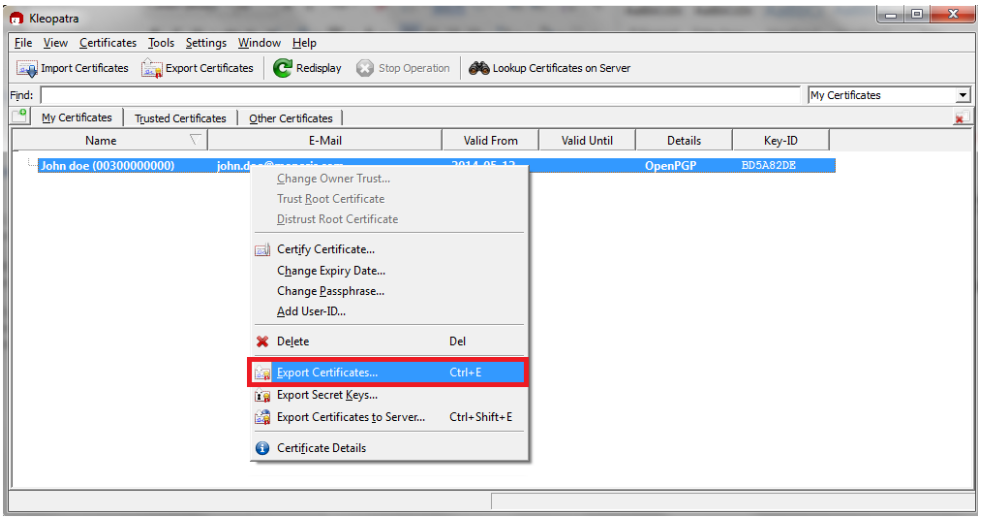
5. Fill in the Full Name, Email, and Comment.

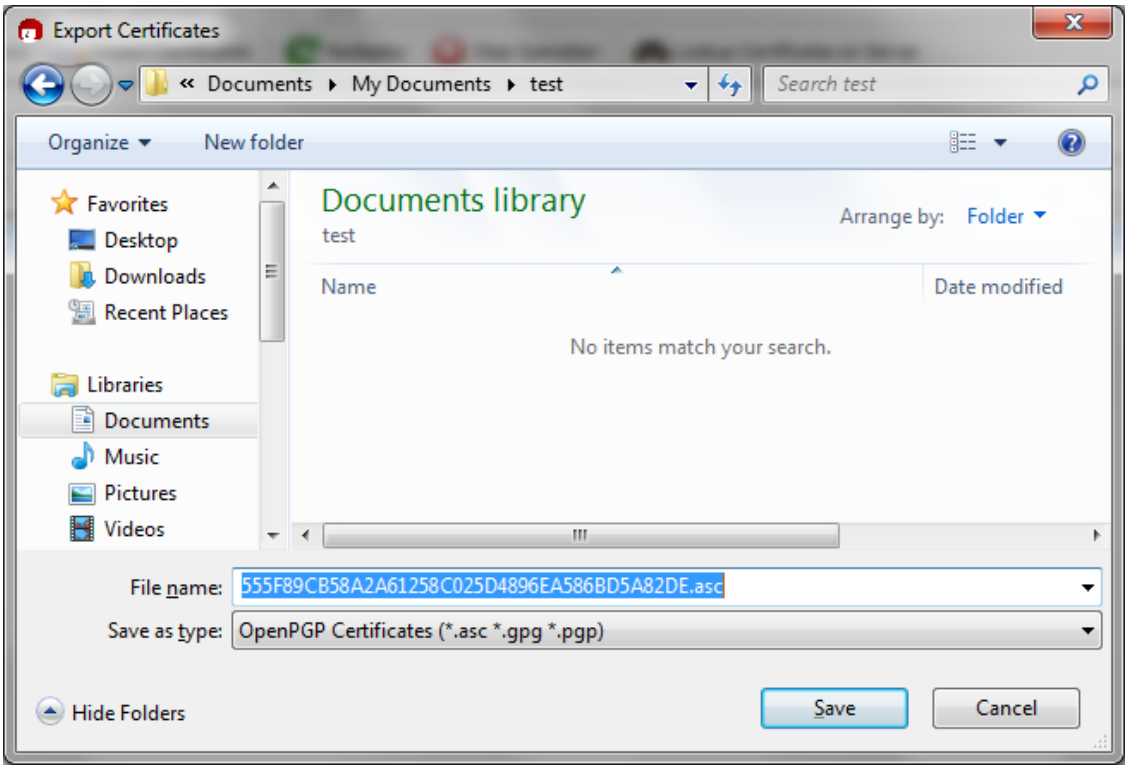
> **NOTE:** The comment field is the most important and the 13 digit Moneris merchant number has to be placed in there.

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



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 E Error Messages

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

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

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

Message: 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