



BE PAYMENT READY

Transaction DTD Merchant Integration Guide

Version: 1.4.2

Copyright © Moneris Solutions, 2021

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

Table of Contents

Getting Help	5
Changes in v1.4.2	6
1 About the Moneris Gateway Transaction DTD	7
1.1 DTD and XML Syntax	7
2 Basic Transaction Set	8
2.1 Process Flow for Basic Transactions	9
2.2 Basic Request DTD	9
2.3 Basic Response DTD	13
2.4 Purchase	14
2.5 Pre-Authorization	19
2.6 Re-Authorization	24
2.7 Pre-Authorization Completion	27
2.8 Force Post	30
2.9 Refund	34
2.10 Independent Refund	37
2.11 Purchase Correction	41
2.12 Card Verification	44
2.13 Batch Close	47
2.14 Open Totals	48
3 Credential on File	50
3.1 About Credential on File	50
3.2 Credential on File Info Object and Variables	50
3.3 Credential on File Transaction Types	51
3.4 Initial Transactions in Credential on File	51
3.5 Credential on File and Converting Temporary Tokens	52
3.6 Card Verification and Credential on File Transactions	52
3.6.1 When to Use Card Verification With COF	53
3.6.2 Credential on File and Vault Add Token	53
3.6.3 Credential on File and Vault Update Credit Card	53
3.6.4 Credential on File and Vault Add Credit Card	53
3.6.5 Credential on File and Recurring Billing	54
3.7 Definition of Request Fields – Credential on File	54
4 Vault Transaction Set	57
4.1 About the Vault Transaction Set	57
4.2 Vault Request DTD	57
4.3 Vault Response DTD	59
4.4 Vault Administrative Transactions	60
4.4.1 Vault Add Credit Card – res_add_cc	60
4.4.2 Vault Update Credit Card – res_update_cc	63
4.4.3 Vault Delete – res_delete	66
4.4.4 Vault Lookup Full – res_lookup_full	68
4.4.5 Vault Lookup Masked – res_lookup_masked	70
4.4.6 Vault Is Corporate Card – res_iscorporatecard	71
4.4.7 Vault Get Expiring – res_get_expiring	72
4.4.8 Vault Temporary Token Add – res_temp_add	74
4.4.9 Vault Add Token – res_add_token	77

4.5	Vault Financial Transactions	80
4.5.1	Customer ID Changes	80
4.5.2	Purchase with Vault – res_purchase_cc	81
4.5.3	Pre-Authorization with Vault – res_preauth_cc	84
4.5.4	Independent Refund with Vault – res_ind_refund_cc	88
4.5.5	Force Post with Vault – res_forcepost_cc	91
4.5.6	Card Verification with Vault – res_card_verification_cc	94
5	3-D Secure 2.0 Transactions	98
5.1	About 3-D Secure 2.0	98
5.1.1	3-D Secure Implementations	98
5.1.2	Out of Scope/Not Supported	99
5.1.3	Version Compatibility	99
5.1.4	Upgrading from 3-D Secure 1.0 to 3-D Secure 2.0	99
5.2	Building Your 3-D Secure 2.0 Integration	99
5.2.1	Activating 3-D Secure Functionality	100
5.2.2	Transaction Flow for 3-D Secure 2.0	100
5.2.3	Mpi2Request Object and 3-D Secure Authentication	101
5.2.4	3-D Secure 2.0 Request DTD	101
5.2.5	3-D Secure 2.0 Response DTD	102
5.3	Implementing Card Lookup Request	102
5.3.1	Card Lookup Request	102
5.3.2	Card Lookup Request – mpiCardLookup	104
5.4	Implementing MPI 3DS Authentication Request	106
5.4.1	MPI 3DS Authentication Request	106
5.5	Handling the Challenge Flow	112
5.5.1	Cavv Lookup Request	112
5.6	Performing the Authorization	114
5.6.1	Purchase with 3-D Secure – cavv_purchase	114
5.6.2	Pre-Authorization with 3-D Secure – cavv_preauth	117
5.7	Testing Your 3-D Secure 2.0 Integration	120
5.8	Moving to Production With 3-D Secure 2.0	121
5.9	3-D Secure 2.0 TransStatus Codes	121
5.10	CAVV Result Codes	122
5.10.1	Visa CAVV Result Codes	122
5.10.2	Mastercard CAVV Result Codes	123
5.10.3	American Express CAVV Result Codes	124
6	Multi-Currency Pricing (MCP)	125
6.1	About Multi-Currency Pricing (MCP)	125
6.2	Methods of Processing MCP Transactions	125
6.3	Multi-Currency Pricing (MCP) Request DTD	126
6.4	Multi-Currency Pricing (MCP) Response DTD	126
6.5	MCP Purchase	126
6.6	MCP Pre-Authorization	131
6.7	MCP Pre-Authorization Completion	136
6.8	MCP Purchase Correction	140
6.9	MCP Refund	143
6.10	MCP Independent Refund	147
6.11	MCP Purchase With Vault	150
6.12	MCP Pre-Authorization With Vault	153
6.13	MCP Independent Refund with Vault	157
6.14	MCP Get Rate	161

6.15 MCP Currency Codes	163
6.16 MCP Error Codes	169
7 Apple Pay Token Transactions	170
7.1 About Apple Pay Token Transactions	170
7.2 Apple Pay Token Request DTD	170
7.3 Apple Pay Token Purchase	171
7.4 Apple Pay Token Pre-Authorization	175
8 Recurring Billing	180
8.1 About Recurring Billing	180
8.2 Purchase with Recurring Billing	180
8.3 Recurring Billing Update	186
8.4 Recurring Billing Response Fields and Codes	189
8.5 Credential on File and Recurring Billing	191
9 Customer Information	192
9.1 Customer Information Object	192
9.2 Customer Info Object – Billing Information	193
9.3 Customer Info Object – Shipping Information	194
9.4 Customer Information Object – Items	196
10 e-Fraud Tools	197
10.1 Address Verification Service (AVS)	197
10.1.1 About Address Verification Service (AVS)	197
10.1.2 AVS Information Object	198
10.1.3 AVS Response Codes	198
10.2 Card Validation Digits (CVD)	202
10.2.1 About Card Validation Digits (CVD)	202
10.2.2 Transactions Where CVD Is Required	203
10.2.3 CVD Information Object	203
10.2.4 CVD Result Codes	204
11 Definition of Request Fields	206
11.1 Definition of Request Fields – Admin Transactions	210
11.2 Definition of Request Fields – Vault	210
11.3 Definition of Request Fields – 3-D Secure 2.0	211
11.4 Definition of Request Fields – Information Objects	215
11.5 Definition of Request Fields – Credential on File	215
11.6 Definition of Request Fields – Apple Pay Token	217
11.7 Definition of Request Fields – Recurring Billing	219
11.8 Definition of Request Fields – AVS Info Object	220
11.9 Definition of Request Fields – CVD Info Object	221

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	If you are already working with an integration spe- cialist and need technical development assistance, contact our eProducts Technical Consultants: 1-866-319-7450 eproducts@moneris.com	If your application is already live and you need production sup- port, 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 v1.4.2

- Corrected limits for the request field **start date**

Changes in v1.4.1

- added topic DTD and XML Syntax
- added changes in version topic

1 About the Moneris Gateway Transaction DTD

The Moneris Gateway supports processing of credit card and debit card transactions in XML format over the HTTPS protocol.

This document contains detailed information on the request and response transaction requirements of the XML format. When creating custom APIs, these requirements must be met in order for transactions to be sent to Moneris Gateway in the proper format.

1.1 DTD and XML Syntax

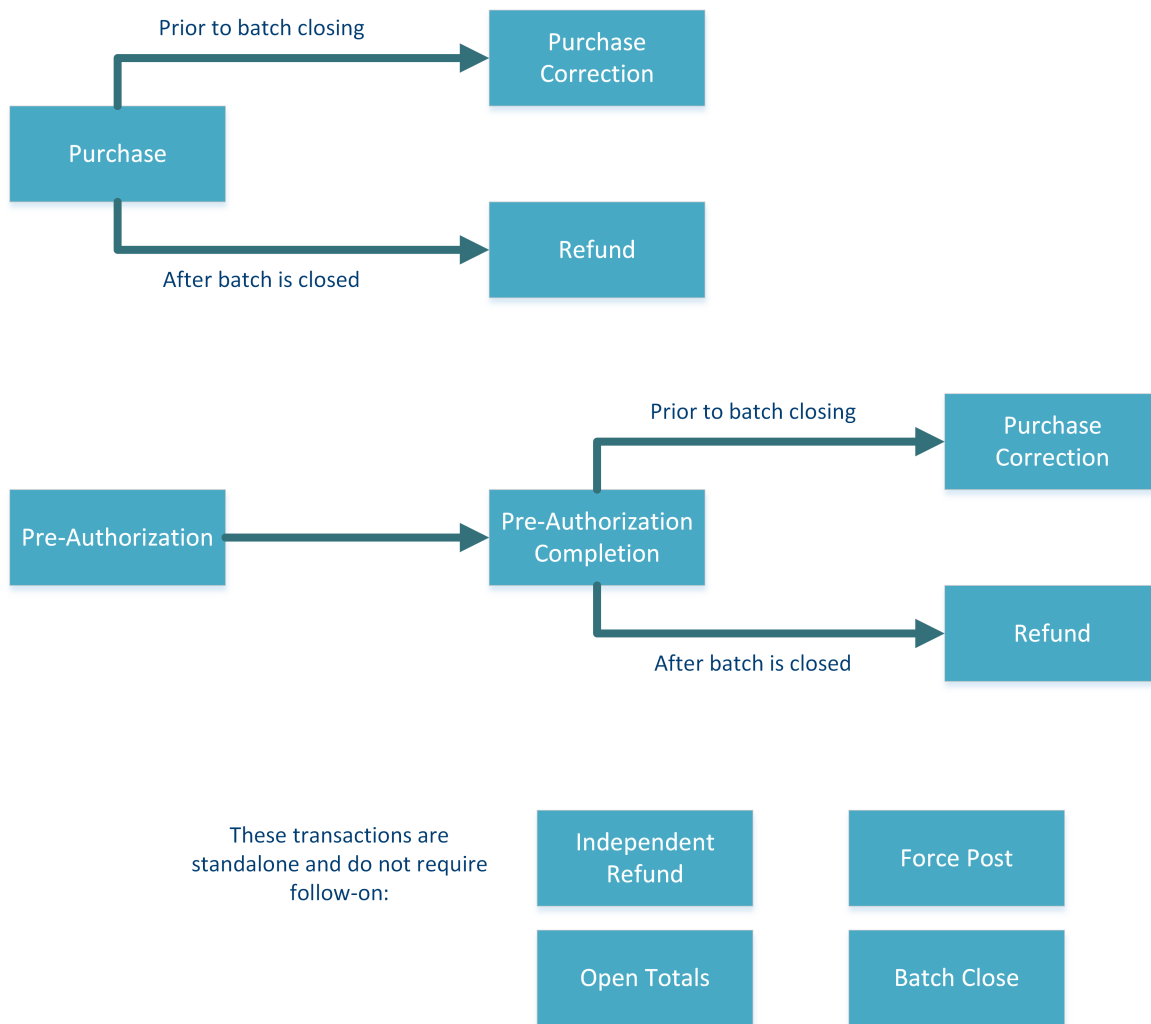
The DTD in this document provides a structural map for constructing the XML code for transactions on the Moneris Gateway.

When coding the XML, self-closing XML tags (i.e., `<tag/>`) should not be used. Open/close tags should always be employed (i.e., `<tag></tag>`).

2 Basic Transaction Set

- 2.1 Process Flow for Basic Transactions
- 2.2 Basic Request DTD
- 2.3 Basic Response DTD
- 2.4 Purchase
- 2.5 Pre-Authorization
- 2.6 Re-Authorization
- 2.7 Pre-Authorization Completion
- 2.8 Force Post
- 2.9 Refund
- 2.10 Independent Refund
- 2.11 Purchase Correction
- 2.12 Card Verification
- 2.13 Batch Close
- 2.14 Open Totals

2.1 Process Flow for Basic Transactions



2.2 Basic Request DTD

```
<!-- The Request DTD CA -->
<!-- Main Elements -->
<!ELEMENT request (store_id, api_token, status_check?, (purchase | refund | ind_refund |
preauth | completion | purchasecorrection |
forcepost | reauth | card_verification | idebit_purchase | idebit_refund | cavv_preauth |
cavv_purchase |
mcp_completion | mcp_ind_refund | mcp_preauth | mcp_purchase | mcp_purchasecorrection | mcp_
refund | mcp_res_ind_refund_cc | mcp_res_preauth_cc | mcp_res_purchase_cc | mcp_get_rate |
batchclose | opentotals | recur_update | applepay_token_preauth | applepay_token_purchase |
googlepay_preauth | googlepay_purchase))>
<!ELEMENT store_id (#PCDATA)>
<!ELEMENT api_token (#PCDATA)>
<!ELEMENT status_check (#PCDATA)>

<!--The following are the basic credit card transactions -->
<!-- purchase -->
<!ELEMENT purchase (order_id, cust_id?, amount, pan, expdate, crypt_type, dynamic_
descriptor?, cust_info?, avs_info?, cvd_info?, recur?, cof_info? )>
<!-- refund -->
<!ELEMENT refund (order_id, amount, txn_number, crypt_type)>
```

```

<!ELEMENT ind_refund (order_id, cust_id?, amount, pan, expdate, crypt_type, dynamic_
descriptor?)>
<!ELEMENT preauth (order_id, cust_id?, amount, pan, expdate, crypt_type, dynamic_descriptor?,
cust_info?, avs_info?, cvd_info?, cof_info?)>
<!ELEMENT completion (order_id, comp_amount, txn_number, crypt_type, ship_indicator?)>
<!ELEMENT purchasecorrection (order_id, txn_number, crypt_type, ship_indicator?)>
<!ELEMENT forcepost (order_id, cust_id?, amount, pan, expdate, crypt_type, auth_code,
dynamic_descriptor?)>
<!ELEMENT card_verification (order_id, cust_id?, pan, expdate, crypt_type, avs_info?, cvd_
info?, cof_info?)>
<!ELEMENT reauth (order_id, cust_id?, orig_order_id, txn_number, amount, crypt_type)>
<!--The following are the Interac Online transactions -->
<!ELEMENT idebit_purchase (order_id, cust_id?, amount, idebit_track2) >
<!ELEMENT idebit_refund (order_id, amount, txn_number)>
<!--The following are for Verified by Visa and/or MasterCard SecureCode transactions -->
<!--NOTE: To assist in any chargeback investigations, it is recommended to store the 'XID'
(order_id) returned from the MPI for future reference. -->
<!ELEMENT cavv_preauth (order_id, cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threads_version?, threads_server_trans_id?, cust_
info?, avs_info?, cvd_info?, cof_info?)>
<!ELEMENT cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threads_version?, threads_server_trans_id?, cust_
info?, avs_info?, cvd_info?, recur?, cof_info?)>

<!--The following are the Multi-currency transactions (MCP) -->
<!ELEMENT mcp_completion (order_id, txn_number, crypt_type, cust_id, dynamic_descriptor?,
ship_indicator?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?) >
<!ELEMENT mcp_ind_refund (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_preauth (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
<!ELEMENT mcp_purchase (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
<!ELEMENT mcp_purchasecorrection (order_id, txn_number, crypt_type, cust_id)>
<!ELEMENT mcp_refund (order_id, amount, txn_number, crypt_type, cust_id, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_ind_refund_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_preauth_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_purchase_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_get_rate (mcp_version, rate_txn_type, rate_info)>

<!--The following are general administrative transactions -->
<!ELEMENT batchclose (ecr_number)>
<!ELEMENT opentotals (ecr_number)>
<!ELEMENT recur_update (order_id, cust_id?, pan?, expdate?, recur_amount?, add_num_recur?,
total_num_recur?, hold?, terminate?, cof_info?)>

<!--The following are the Wallet Transactions (ApplePay and GooglePay) -->
<!ELEMENT applepay_token_purchase (order_id, cust_id?, amount, displayName, network, version,
data, signature, header, type, dynamic_descriptor?, token_originator?)>
<!ELEMENT applepay_token_preauth (order_id, cust_id?, amount, displayName, network, version,
data, signature, header, type, dynamic_descriptor?, token_originator?)>
<!ELEMENT googlepay_purchase (order_id, cust_id?, amount, network, payment_token, dynamic_
descriptor?)>
<!ELEMENT googlepay_preauth (order_id, cust_id?, amount, network, payment_token, dynamic_
descriptor?)>

<!--The following are the 3DS 2.0 transactions -->

```

```
<!ELEMENT Mpi2Request (store_id, api_token, (card_lookup | threeds_authentication | cavv_
lookup))>
<!ELEMENT card_lookup (order_id, (data_key | pan), notification_url)>
<!ELEMENT threeds_authentication (order_id, cardholder_name, (data_key | pan), expdate,
amount, currency?, threeds_completion_ind, request_type, notification_url, purchase_date,
challenge_windowsize, bill_address1?, bill_province?, bill_city?, bill_postal_code?, bill_
country?, ship_address1?, ship_province?, ship_city?, ship_postal_code?, ship_country?,
browser_useragent, browser_java_enabled, browser_screen_height, browser_screen_width,
browser_language, email?, request_challenge?)>
<!ELEMENT cavv_lookup (cres)>

<!-- start standard -->
<!ELEMENT order_id (#PCDATA)>
<!ELEMENT orig_order_id (#PCDATA)>
<!ELEMENT cust_id (#PCDATA)>
<!ELEMENT txn_number (#PCDATA)>
<!ELEMENT crypt_type (#PCDATA)>
<!ELEMENT auth_code (#PCDATA)>
<!ELEMENT cavv (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT comp_amount (#PCDATA)>
<!ELEMENT pan (#PCDATA)>
<!ELEMENT idebit_track2 (#PCDATA)>
<!ELEMENT expdate (#PCDATA)>
<!ELEMENT ecr_number (#PCDATA)>
<!ELEMENT dynamic_descriptor (#PCDATA)>
<!ELEMENT add_num_rekurs (#PCDATA)>
<!ELEMENT total_num_rekurs (#PCDATA)>
<!ELEMENT hold (#PCDATA)>
<!ELEMENT terminate (#PCDATA)>
<!ELEMENT ship_indicator (#PCDATA)>
<!ELEMENT wallet_indicator (#PCDATA)>
<!ELEMENT market_indicator (#PCDATA)>
<!-- start Cust Info -->
<!ELEMENT cust_info (billing, shipping, email, instructions, item+)>
<!ELEMENT billing (first_name, last_name, company_name, address, city, province, postal_code,
country, phone_number, fax, tax1, tax2, tax3, shipping_cost)>
<!ELEMENT shipping (first_name, last_name, company_name, address, city, province, postal_
code, country, phone_number, fax, tax1, tax2, tax3, shipping_cost)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT company_name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT province (#PCDATA)>
<!ELEMENT postal_code (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT tax1 (#PCDATA)>
<!ELEMENT tax2 (#PCDATA)>
<!ELEMENT tax3 (#PCDATA)>
<!ELEMENT shipping_cost (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT instructions (#PCDATA)>
<!ELEMENT item (name, quantity, product_code, extended_amount)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
<!ELEMENT product_code (#PCDATA)>
<!ELEMENT extended_amount (#PCDATA)>
<!-- start AVS -->
<!ELEMENT avs_info (avs_street_number, avs_street_name, avs_zipcode, avs_email?, avs_
hostname?, avs_browser?, avs_shiptocountry?, avs_shipmethod?, avs_merchprodsku?, avs_custip?,
avs_custphone?)>
<!ELEMENT avs_street_number (#PCDATA)>
```

```

<!ELEMENT avs_street_name (#PCDATA)>
<!ELEMENT avs_zipcode (#PCDATA)>
<!ELEMENT avs_email (#PCDATA)>
<!ELEMENT avs_hostname (#PCDATA)>
<!ELEMENT avs_browser (#PCDATA)>
<!ELEMENT avs_shiptocountry (#PCDATA)>
<!ELEMENT avs_shipmethod (#PCDATA)>
<!ELEMENT avs_merchprodsku (#PCDATA)>
<!ELEMENT avs_custip (#PCDATA)>
<!ELEMENT avs_custphone (#PCDATA)>
<!-- start CVD -->
<!ELEMENT cvd_info (cvd_indicator, cvd_value)>
<!ELEMENT cvd_indicator (#PCDATA)>
<!ELEMENT cvd_value (#PCDATA)>
<!-- start Recur -->
<!ELEMENT recur (recur_unit, start_now, start_date, num_recur, period, recur_amount)>
<!ELEMENT recur_unit (#PCDATA)>
<!ELEMENT start_now (#PCDATA)>
<!ELEMENT start_date (#PCDATA)>
<!ELEMENT num_recur (#PCDATA)>
<!ELEMENT period (#PCDATA)>
<!ELEMENT recur_amount (#PCDATA)>
<!-- start COF -->
<!ELEMENT cof_info (payment_indicator, payment_information, issuer_id)>
<!ELEMENT payment_indicator (#PCDATA)>
<!ELEMENT payment_information (#PCDATA)>
<!ELEMENT issuer_id (#PCDATA)>
<!-- start MCP specific fields-->
<!ELEMENT rate_info (rate+)>
<!ELEMENT rate ((cardholder_amount | merchant_settlement_amount), cardholder_currency_code)>
<!ELEMENT mcp_version (#PCDATA)>
<!ELEMENT rate_txn_type (#PCDATA)>
<!ELEMENT cardholder_amount (#PCDATA)>
<!ELEMENT merchant_settlement_amount (#PCDATA)>
<!ELEMENT cardholder_currency_code (#PCDATA)>
<!ELEMENT mcp_rate_token (#PCDATA)>
<!ELEMENT cm_id (#PCDATA)>
<!-- start Wallet specific fields -->
<!ELEMENT token_originator (store_id, api_token)>
<!ELEMENT displayName (#PCDATA)>
<!ELEMENT network (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT header (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!-- start 3DS 2.0 specific fields -->
<!ELEMENT threads_version (#PCDATA)>
<!ELEMENT threads_server_trans_id (#PCDATA)>
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT notification_url (#PCDATA)>
<!ELEMENT cardholder_name (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT threads_completion_ind (#PCDATA)>
<!ELEMENT request_type (#PCDATA)>
<!ELEMENT purchase_date (#PCDATA)>
<!ELEMENT challenge_window_size (#PCDATA)>
<!ELEMENT bill_address1 (#PCDATA)>
<!ELEMENT bill_province (#PCDATA)>
<!ELEMENT bill_city (#PCDATA)>
<!ELEMENT bill_postal_code (#PCDATA)>
<!ELEMENT bill_country (#PCDATA)>
<!ELEMENT ship_address1 (#PCDATA)>
<!ELEMENT ship_province (#PCDATA)>
<!ELEMENT ship_city (#PCDATA)>
<!ELEMENT ship_postal_code (#PCDATA)>

```

```
<!ELEMENT ship_country (#PCDATA)>
<!ELEMENT browser_useragent (#PCDATA)>
<!ELEMENT browser_java_enabled (#PCDATA)>
<!ELEMENT browser_screen_height (#PCDATA)>
<!ELEMENT browser_screen_width (#PCDATA)>
<!ELEMENT browser_language (#PCDATA)>
<!ELEMENT request_challenge (#PCDATA)>
<!ELEMENT cres (#PCDATA)>
```

2.3 Basic Response DTD

```
<!-- The Response DTD CA-->
<!-- Main Elements -->
<!ELEMENT response (receipt+)>
<!ELEMENT Mpi2Response (receipt)>
<!ELEMENT receipt ((ReceiptId, ReferenceNum, ResponseCode, ISO, AuthCode, TransTime,
TransDate, TransType, Complete,
Message, TransAmount, CardType, TransID, TimedOut, BankTotals, Ticket, RecurSuccess?,
CvdResultCode?, AvsResultCode?,
CavvResultCode?, ITDResponse?, StatusCode?, StatusMessage?, RecurUpdateSuccess?,
NextRecurDate?, RecurEndDate?, IsVisaDebit,
MCPRateToken?, RateTxnType?, (Rate+)?, RateInqStartTime?, RateInqEndTime?,
RateValidityStartTime?, RateValidityEndTime?, RateValidityPeriod?,
CardholderCurrencyCode?, CardholderAmount?, MerchantSettlementCurrency?,
MerchantSettlementAmount?, MCPRate?, MCPErrorStatusCode?, MCPErrorMessage?) |
(MessageType?, ResponseCode, Message, ReceiptId, ThreeDSMethodURL?, ThreeDSMethodData?,
ChallengeURL?, ChallengeData?, ChallengeCompletionIndicator?, TransStatus?,
ThreeDSServerTransId, ECI?, Cavv?))>
<!ELEMENT ReceiptId (#PCDATA)>
<!ELEMENT ReferenceNum (#PCDATA)>
<!ELEMENT ResponseCode (#PCDATA)>
<!ELEMENT ISO (#PCDATA)>
<!ELEMENT AuthCode (#PCDATA)>
<!ELEMENT TransTime (#PCDATA)>
<!ELEMENT TransDate (#PCDATA)>
<!ELEMENT TransType (#PCDATA)>
<!ELEMENT Complete (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
<!ELEMENT TransAmount (#PCDATA)>
<!ELEMENT CardType (#PCDATA)>
<!ELEMENT TransID (#PCDATA)>
<!ELEMENT TimedOut (#PCDATA)>
<!ELEMENT BankTotals (ECR)>
<!ELEMENT Ticket (#PCDATA)>
<!ELEMENT CvdResultCode (#PCDATA)>
<!ELEMENT AvsResultCode (#PCDATA)>
<!ELEMENT RecurSuccess (#PCDATA)>
<!ELEMENT IsVisaDebit (#PCDATA)>
<!ELEMENT IssuerId (#PCDATA)>
<!-- The following is only applicable if for CAVV validation transactions -->
<!ELEMENT CavvResultCode (#PCDATA)>
<!-- The following is only applicable for Amex and JCB ITD Validation -->
<!ELEMENT ITDResponse (#PCDATA)>
<!-- The following are only applicable if status_check is set to true in the request -->
<!ELEMENT StatusCode (#PCDATA)>
<!ELEMENT StatusMessage (#PCDATA)>
<!-- The following are only applicable to the recur update transaction -->
<!ELEMENT RecurUpdateSuccess (#PCDATA)>
<!ELEMENT NextRecurDate (#PCDATA)>
<!ELEMENT RecurEndDate (#PCDATA)>
<!-- The following are only applicable in a batch close or open totals transaction -->
<!ELEMENT ECR (term_id, closed, Card+)>
<!ELEMENT Card (CardType, (Purchase | Refund | Correction)+)>
<!ELEMENT term_id (#PCDATA)>
```

```

<!ELEMENT closed (#PCDATA)>
<!ELEMENT Purchase (Count, Amount)>
<!ELEMENT Refund (Count, Amount)>
<!ELEMENT Correction (Count, Amount)>
<!ELEMENT Count (#PCDATA)>
<!ELEMENT Amount (#PCDATA)>
<!-- The following are only applicable to Multi-currency transactions (MCP) -->
<!ELEMENT Rate (CardholderCurrencyCode, CardholderAmount, MerchantSettlementCurrency,
MerchantSettlementAmount, MCPRate, MCPErrorStatusCode, MCPErrorMessage)>
<!ELEMENT MCPRateToken (#PCDATA)>
<!ELEMENT RateTxnType (#PCDATA)>
<!ELEMENT RateInqStartTime (#PCDATA)>
<!ELEMENT RateInqEndTime (#PCDATA)>
<!ELEMENT RateValidityStartTime (#PCDATA)>
<!ELEMENT RateValidityEndTime (#PCDATA)>
<!ELEMENT RateValidityPeriod (#PCDATA)>
<!ELEMENT CardholderCurrencyCode (#PCDATA)>
<!ELEMENT CardholderAmount (#PCDATA)>
<!ELEMENT MerchantSettlementCurrency (#PCDATA)>
<!ELEMENT MerchantSettlementAmount (#PCDATA)>
<!ELEMENT MCPRate (#PCDATA)>
<!ELEMENT MCPErrorStatusCode (#PCDATA)>
<!ELEMENT MCPErrorMessage (#PCDATA)>
<!-- The following are only applicable to 3DS 2.0 transactions -->
<!ELEMENT MessageType (#PCDATA)>
<!ELEMENT ThreeDSMethodURL (#PCDATA)>
<!ELEMENT ThreeDSMethodData (#PCDATA)>
<!ELEMENT ChallengeURL (#PCDATA)>
<!ELEMENT ChallengeData (#PCDATA)>
<!ELEMENT ChallengeCompletionIndicator (#PCDATA)>
<!ELEMENT TransStatus (#PCDATA)>
<!ELEMENT ThreeDSSTransId (#PCDATA)>
<!ELEMENT ECI (#PCDATA)>
<!ELEMENT Cavv (#PCDATA)>

```

2.4 Purchase

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

XML transaction object

<purchase>

Purchase transaction object definition

```

<!ELEMENT purchase (order_id?, cust_id?, amount, pan, expdate, crypt_type,
dynamic_descriptor?, cust_info?, avs_info?, cvd_info?, recur?, cof_info? )>

```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token	String	Unique alphanumeric string assigned

Variable Name	Type and Limits	Description
<api_token>	N/A	<p>by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Purchase transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>

Variable Name	Type and Limits	Description
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
credit card number <pan >	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. <div>NOTE: This is the reverse of the MMY date format that is presented on the card.</div>
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator , as follows:

Variable Name	Type and Limits	Description
		<p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Purchase transaction request fields – Optional

Variable Name	Type and Limits	Description
<p>customer ID</p> <p><cust_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
<p>dynamic descriptor</p> <p><dynamic_descriptor></p>	<p><i>String</i></p> <p>20-character alphanumeric</p> <p>total of 22 characters including your merchant name and separator</p> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <p>NOTE: The 22-character maximum limit must take the "/" into account as one of the characters</p>
<p>wallet indicator</p> <p><wallet_indicator></p>	<p><i>String</i></p> <p>3-character alphanumeric</p>	<p>Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass.</p> <p>Required for Apple Pay and Google Pay transactions whereby you are</p>

Variable Name	Type and Limits	Description
		<p>using your own API to decrypt the payload</p> <p>Possible values:</p> <p>APP – Apple Pay In-App</p> <p>APW – Apple Pay on the Web</p> <p>GPP – Google Pay™ In-App</p> <p>GPW – Google Pay™ Web</p> <p>VCO – Visa Checkout</p> <p>MMP – Mastercard MasterPass</p> <div> NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. </div>
Customer Information <cust_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur> <div> For information on request </div>	<i>Object</i> N/A	Contains fields related to Recurring Billing

Variable Name	Type and Limits	Description
fields for this object, see xrefHere		
Credential on File Information <cof_info>	Object N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
For information on request fields for this object, see xrefHere		

2.5 Pre-Authorization

Verifies and locks funds on the customer's credit card. The funds are locked for a specified amount of time based on the card issuer.

To retrieve the funds that have been locked by a Pre-Authorization transaction so that they may be settled in the merchant's account, a Pre-Authorization Completion transaction must be performed. A Pre-Authorization transaction may only be "completed" once.

XML transaction object

<preauth>

Pre-Authorization transaction object definition

```
<!ELEMENT preauth (order_id, cust_id?, amount, pan, expdate, crypt_type,
dynamic_descriptor?, cust_info?, avs_info?, cvd_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.-

Variable Name	Type and Limits	Description
		moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Pre-Authorization transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>

Variable Name	Type and Limits	Description
<div>EXAMPLE: 1234567.89</div>		
credit card number <pan>	String max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	String 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. <div>NOTE: This is the reverse of the MMYM date format that is presented on the card.</div>
electronic commerce indicator <crypt_type>	String 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator , as follows: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values

Variable Name	Type and Limits	Description
		for e-commerce indicator: 1, 5, 6 or 7

Pre-Authorization transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div>
wallet indicator <wallet_indicator>	<i>String</i> 3-character alphanumeric	Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass. Required for Apple Pay and Google Pay transactions whereby you are using your own API to decrypt the payload Possible values: APP – Apple Pay In-App APW – Apple Pay on the Web GPP – Google Pay™ In-App

Variable Name	Type and Limits	Description
		GPW – Google Pay™ Web VCO – Visa Checkout MMP – Mastercard MasterPass <div> NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. </div>
Customer Information <cust_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur> <div> For information on request fields for this object, see xrefHere </div>	<i>Object</i> N/A	Contains fields related to Recurring Billing
Credential on File Information <cof_info> <div> For information on request </div>	<i>Object</i> N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

Variable Name	Type and Limits	Description
fields for this object, see xrefHere		

2.6 Re-Authorization

If a Pre-Authorization transaction has already taken place, and not all the locked funds were released by a Completion transaction, a Re-Authorization allows you to lock the remaining funds so that they can be released by another Completion transaction in the future.

Re-Authorization is necessary because funds that have been locked by a Pre-Authorization transaction can only be released by a Completion transaction one time. If the Completion amount is less than the Pre-Authorization amount, the remaining money cannot be "completed".

XML transaction object

<reauth>

TransactionTopicName transaction object definition

```
<!ELEMENT reauth (order_id, cust_id?, orig_order_id, txn_number, amount, crypt_type)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

TransactionTopicName transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <xmlvariableHere>	<i>String</i> 50-character alphanumeric- Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
original order ID <orig_order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	<p>Order ID from the original Pre-Authorization transaction, used as a reference to retrieve the original payment details</p>
transaction number <txn_number>	<i>String</i> 255-character, alpha- numeric, hyphens or under- scores variable length	<p>Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)</p> <p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: ref-</p>

Variable Name	Type and Limits	Description
		<p>erences a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p>
<p>amount</p> <p><amount></p>	<p><i>String</i></p> <p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p> <p>EXAMPLE: 1234567.89</p>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
<p>electronic commerce indicator</p> <p><crypt_type></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <p>1 – Mail Order / Telephone Order—Single</p> <p>2 – Mail Order / Telephone Order—Recurring</p> <p>3 – Mail Order / Telephone Order—Installment</p> <p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Variable Name	Type and Limits	Description
		<p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

TransactionTopicName transaction request fields – Optional

Variable Name	Type and Limits	Description
<p>customer ID</p> <p><cust_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>

2.7 Pre-Authorization Completion

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

XML transaction object

<completion>

Pre-Authorization Completion transaction object definition

```
<!ELEMENT completion (order_id, comp_amount, txn_number, crypt_type, ship_indicator?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
<p>store ID</p> <p><store_id></p>	<p><i>String</i></p> <p>N/A</p>	<p>Unique identifier provided by Moneris upon merchant account setup</p>
<p>API token</p> <p><api_token></p>	<p><i>String</i></p> <p>N/A</p>	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p>

Variable Name	Type and Limits	Description
		<p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
<p>status check</p> <p><status_check></p>	<p><i>Boolean</i></p> <p>true/false</p>	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Pre-Authorization Completion transaction request fields – Required

Variable Name	Type and Limits	Description
<p>order ID</p> <p><order_id></p>	<p><i>String</i></p> <p>50-character alphanumeric-A-Z 0-9 _ - : . @ spaces</p>	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
<p>completion amount</p> <p><comp_amount></p>	<p><i>String</i></p> <p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p>	<p>Dollar amount of a Pre-Authorization Completion transaction, which may differ from the original amount authorized in the Pre-Authorization</p>

Variable Name	Type and Limits	Description
<div>EXAMPLE: 1234567.89</div>		
transaction number <txn_number>	<i>String</i> 255-character, alpha-numeric, hyphens or underscores variable length	Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund) This value is returned in the response of the original transaction Pre-Authorization Completion: references a Pre-Authorization Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator , as follows: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values

Variable Name	Type and Limits	Description
		for e-commerce indicator: 1 or 7
		if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7

Pre-Authorization Completion transaction request fields – Optional

Variable Name	Type and Limits	Description
shipping indicator <ship_indicator>	<i>String</i> 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

2.8 Force Post

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

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

XML transaction object

<forcepost>

Force Post transaction object definition

```
<!ELEMENT forcepost (order_id, cust_id?, amount, pan, expdate, crypt_type, auth_code, dynamic_descriptor?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Force Post transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may

Variable Name	Type and Limits	Description
		<p>have the same order ID.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<p><i>String</i></p> <p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p> <p>EXAMPLE: 1234567.89</p>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
credit card number <pan>	<p><i>String</i></p> <p>max 20-character alphanumeric</p>	<p>Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.</p>
expiry date <expdate>	<p><i>String</i></p> <p>4-character alphanumeric</p> <p>YYMM</p>	<p>Expiry date of the credit card, in YYMM format.</p> <p>NOTE: This is the reverse of the MMY date format that is presented on the card.</p>
electronic commerce indicator <crypt_type>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant

Variable Name	Type and Limits	Description
		<p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
authorization code <auth_code>	String 8-character alphanumeric	An authorization code required to carry out a Force Post; provided in the transaction response from the issuing bank

Force Post transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	String 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated

Variable Name	Type and Limits	Description
---------------	-----------------	-------------

NOTE: The 22-character maximum limit must take the "/" into account as one of the characters

2.9 Refund

Restores all or part of the funds from a Purchase, Pre-Authorization Completion or Force Post transaction to the cardholder's card.

Unlike a Purchase Correction, there is a record of both the initial charge and the refund on the cardholder's statement.

For processing refunds on a different card than the one used in the original transaction, the Independent Refund transaction should be used instead.

XML transaction object

<refund>

Refund transaction object definition

<!ELEMENT refund (order_id, amount, txn_number, crypt_type, cust_id?, dynamic_descriptor?)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alpha-numeric-A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
transaction number <txn_number>	<i>String</i> 255-character, alpha-numeric, hyphens or underscores	Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)

Variable Name	Type and Limits	Description
	variable length	<p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p>
<p>electronic commerce indicator</p> <p><crypt_type></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7

Refund transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div>

2.10 Independent Refund

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

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

XML transaction object

```
<ind_refund>
```

Independent Refund transaction object definition

```
<!ELEMENT ind_refund (order_id, cust_id?, amount, pan, expdate, crypt_type, dynamic_descriptor?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Independent Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may

Variable Name	Type and Limits	Description
		<p>have the same order ID.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<p><i>String</i></p> <p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p> <p>EXAMPLE: 1234567.89</p>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
credit card number <pan>	<p><i>String</i></p> <p>max 20-character alphanumeric</p>	<p>Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.</p>
expiry date <expdate>	<p><i>String</i></p> <p>4-character alphanumeric</p> <p>YYMM</p>	<p>Expiry date of the credit card, in YYMM format.</p> <p>NOTE: This is the reverse of the MMY date format that is presented on the card.</p>
electronic commerce indicator <crypt_type>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant

Variable Name	Type and Limits	Description
		<p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Independent Refund transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
dynamic descriptor <dynamic_descriptor>	<p><i>String</i></p> <p>20-character alphanumeric</p> <p>total of 22 characters including your merchant name and separator</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> <p>NOTE: The 22-character maximum limit must take the "/" into account as one of the characters</p> </div>

2.11 Purchase Correction

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

This transaction can be used against a Purchase or Pre-Authorization Completion transaction that occurred same day provided that the batch containing the original transaction remains open. When using the automated closing feature, Batch Close occurs daily between 10 and 11 pm Eastern Time.

XML transaction object

```
<purchasecorrection>
```

Purchase Correction transaction object definition

```
<!ELEMENT purchasecorrection (order_id, txn_number, crypt_type)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p>

Variable Name	Type and Limits	Description
		<p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p>

Purchase Correction transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alpha-numeric-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
transaction number <txn_number>	<i>String</i> 255-character, alpha-numeric, hyphens or underscores variable length	<p>Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)</p> <p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p>
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <p>1 – Mail Order / Telephone Order—Single</p> <p>2 – Mail Order / Telephone Order—Recurring</p> <p>3 – Mail Order / Telephone Order—Instalment</p> <p>4 – Mail Order / Telephone Order—Unknown classification</p>

Variable Name	Type and Limits	Description
		<p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Purchase Correction transaction request fields – Optional

Variable Name	Type and Limits	Description
<p>shipping indicator</p> <p><ship_indicator></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Used to identify completion transactions that require multiple shipments, also referred to as multiple completions</p> <p>By default, if shipping indicator is not sent, the Pre-Authorization Completion is listed as final</p> <p>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</p> <p>Possible values:</p> <p>P – Partial</p> <p>F – Final</p>

2.12 Card Verification

Verifies the validity of the credit card, expiry date and any additional details (such as the Card Verification Digits or Address Verification details). It does not verify the available amount or lock any funds on the credit card.

XML transaction object

```
<card_verification>
```

Card Verification transaction object definition

```
<!ELEMENT card_verification (order_id, cust_id?, pan, expdate, crypt_type, avs_info?, cvd_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original trans- </div>

Variable Name	Type and Limits	Description
		<div>action request; if the status check request times out, do not send again, as additional investigation is required</div>

Card Verification transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alpha-numeric-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
credit card number <pan>	<i>String</i> max 20-character alpha-numeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. <div> NOTE: This is the reverse of the MMYD date format that is presented on the card. </div>
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce trans-

Variable Name	Type and Limits	Description
		<p>action (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Card Verification transaction request fields – Optional

Variable Name	Type and Limits	Description
<p>customer ID</p> <p><cust_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
<p>AVS Information</p> <p><avs_info></p> <div> <p>For information on request fields for this object, see xrefHere</p> </div>	<p><i>Object</i></p> <p>N/A</p>	<p>Contains fields applying to the Address Verification Service (AVS) e-fraud tool</p>
<p>CVD Information</p> <p><cvd_info></p> <div> <p>For information on request fields for this object, see xrefHere</p> </div>	<p><i>Object</i></p> <p>N/A</p>	<p>Contains fields related to the Card Validation Digits e-fraud tool</p>

Variable Name	Type and Limits	Description
Credential on File Information <cof_info>	Object N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
<div>For information on request fields for this object, see xrefHere</div>		

2.13 Batch Close

Takes the funds from all Purchase, Completion, Refund and Force Post transactions so that they will be deposited or debited the following business day.

For funds to be deposited the following business day, the batch must close before 11 pm Eastern Time.

XML transaction object

<batchclose>

Batch Close transaction object definition

<!ELEMENT batchclose (ecr_number)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

Batch Close transaction request fields – Required

Variable Name	Type and Limits	Description
electronic cash register (ECR) number <ecr_number>	<i>String</i> N/A	Identification number assigned to a particular electronic cash register; provided by Moneris

2.14 Open Totals

Returns the details about the currently open batch.

Similar to the Batch Close; the difference is that it does not close the batch for settlement.

XML transaction object

<opentotals>

Open Totals transaction object definition

<!ELEMENT opentotals (ecr_number)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token	<i>String</i>	Unique alphanumeric string assigned

Variable Name	Type and Limits	Description
<api_token>	N/A	<p>by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Open Totals transaction request fields – Required

Variable Name	Type and Limits	Description
electronic cash register (ECR) number <ecr_number>	<i>String</i> N/A	Identification number assigned to a particular electronic cash register; provided by Moneris

3 Credential on File

- 3.1 About Credential on File
- 3.2 Credential on File Info Object and Variables
- 3.3 Credential on File Transaction Types
- 3.4 Initial Transactions in Credential on File
- 3.5 Credential on File and Converting Temporary Tokens
- 3.6 Card Verification and Credential on File Transactions

3.1 About Credential on File

When storing customers' credit card credentials for use in future authorizations, or when using these credentials in subsequent transactions, card brands now require merchants to indicate this in the transaction request.

In the Moneris API, this is handled by the Moneris Gateway via the inclusion of the Credential on File info object and its variables in the transaction request.

While the requirements for handling Credential on File transactions relate to Visa, Mastercard and Discover only, in order to avoid confusion and prevent error, please implement these changes for all card types and the Moneris system will then correctly flow the relevant card data values as appropriate.

NOTE: If either the first transaction or a Card Verification authorization is declined when attempting to store cardholder credentials, those credentials cannot be stored —therefore the merchant must not use the credential for any subsequent transactions.

3.2 Credential on File Info Object and Variables

The Credential on File Info object is nested within the request for the applicable transaction types.

Credential on File Info Object:

cof

Variables in the cof object:

Payment Indicator

Payment Information

Issuer ID

For more information, see 11.5 Definition of Request Fields – Credential on File

3.3 Credential on File Transaction Types

The Credential on File Info object applies to the following transaction types:

- Purchase
- Pre-Authorization
- Purchase with 3-D Secure – cavv_purchase
- Pre-Authorization with 3-D Secure – cavv_preauth
- Purchase with Vault – res_purchase_cc
- Pre-Authorization with Vault – res_preauth_cc
- Card Verification
- Card Verification with Vault – res_card_verification_cc
- Vault Add Credit Card – res_add_cc
- Vault Update Credit Card – res_update_cc
- Recurring Billing transactions

3.4 Initial Transactions in Credential on File

When sending an *initial* transaction with the Credential on File Info object, i.e., a transaction request where the cardholder's credentials are being stored for the *first* time, it is important to understand the following:

- You must send the cardholder's Card Verification Digits (CVD)
- **Issuer ID** will be sent without a value on the initial transaction, because it is received in the response to that initial transaction; for all *subsequent* merchant-initiated transactions and all administrative transactions you send this **Issuer ID**
- The **payment information** field should always be set to a value of 0 on the first transaction
- The **payment indicator** field should be set to the value that is appropriate for the transaction

3.5 Credential on File and Converting Temporary Tokens

In the event you decide to convert a temporary token representing cardholder credentials into a permanent token, these credentials become stored credentials, and therefore it is necessary to send Credential on File information.

For Vault Temporary Token Add transactions where you subsequently decide to convert the temporary token into a permanent token (stored credentials):

1. Send a transaction request that includes the Credential on File Info object to get the Issuer ID; this can be a Card Verification, Purchase or Pre-Authorization request
2. After completing the transaction, send the Vault Add Token request with the Credential on File object in order to convert the temporary token to a permanent one.

For more information about Vault Temporary Token Add transaction, see 4.4.8 Vault Temporary Token Add – res_temp_add.

3.6 Card Verification and Credential on File Transactions

In the absence of a Purchase or Pre-Authorization, a Card Verification transaction is used to get the unique issuer ID value (**issuerid**) that is used in subsequent Credential on File transactions. Issuer ID is a variable included in the nested Credential on File Info object.

For all first-time transactions, including Card Verification transactions, you must also request the cardholder's Card Verification Details (CVD). For more on CVD, see 1 Card Validation Digits (CVD).

The Card Verification request, including the Credential on File Info object, must be sent immediately prior to storing cardholder credentials.

For information about Card Verification, see 2.12 Card Verification.

3.6.1 When to Use Card Verification With COF

If you are not sending a Purchase or Pre-Authorization transaction (i.e., you are not charging the customer immediately), you must use Card Verification (or in the case of Vault Add Token, Card Verification with Vault) first before running the transaction in order to get the Issuer ID.

Transactions this applies to:

Vault Add Credit Card – res_add_cc

Vault Update Credit Card – res_update_cc

Recurring Billing transactions, if:

- the first transaction is set to start on a future date

3.6.2 Credential on File and Vault Add Token

For Vault Add Token transactions:

1. Send Card Verification with Vault transaction request including the Credential on File object to get the Issuer ID
2. Send the Vault Add Token request including the Credential on File object (with Issuer ID only; other fields are not applicable)

3.6.3 Credential on File and Vault Update Credit Card

For Vault Update Credit Card transactions where you are updating the credit card number:

1. Send Card Verification transaction request including the Credential on File object to get the Issuer ID
2. Send the Vault Update Credit Card request including the Credential on File Info object .

3.6.4 Credential on File and Vault Add Credit Card

For Vault Add Credit Card transactions:

1. Send Card Verification transaction request including the Credential on File object to get the Issuer ID
2. Send the Vault Add Credit Card request including the Credential on File Info object

3.6.5 Credential on File and Recurring Billing

NOTE: The value of the **payment indicator** field must be **R** when sending Recurring Billing transactions.

For Recurring Billing transactions which are set to start **immediately**:

1. Send a Purchase transaction request with both the Recurring Billing and Credential on File info objects (with Recurring Billing object field **start now** = true)

For Recurring Billing transactions which are set to start on a **future** date:

1. Send Card Verification transaction request including the Credential on File info object to get the Issuer ID
2. Send Purchase transaction request with the Recur and Credential on File info objects included

For updating a Recurring Billing series where you are updating the card number (does not apply if you are only modifying the schedule or amount in a recurring series):

1. Send Card Verification request including the Credential on File info object to get the Issuer ID
2. Send a Recurring Billing Update transaction

3.7 Definition of Request Fields – Credential on File

Variable Name	Type and Limits	Description
issuer ID <issuer_id>	<i>String</i> 15-character alphanumeric variable length	Unique identifier for the cardholder's stored credentials Sent back in the response from the card brand when processing a Credential on File transaction

Variable Name	Type and Limits	Description
		<p>If the cardholder's credentials are being stored for the first time, and the issuer ID was returned in the response, you must save the issuer ID on your system to use in subsequent Credential on File transactions (applies to merchant-initiated transactions only)</p> <p>The issuer ID must be saved to your systems when returned from Moneris Gateway in the response data, regardless if the value was received or not</p> <p>As a best practice, if the issuer ID is not returned and you received a value of NULL instead, store that value and send it in the subsequent transaction</p>
payment indicator <payment_indicator>	String 1-character alphabetic	<p>Indicates the current or intended use of the credentials</p> <p>Possible values for first transactions:</p> <p>C - unscheduled Credential on File (first transactions only)</p> <p>R - recurring</p> <p>Possible values for subsequent transactions:</p> <p>R - recurring</p> <p>U - unscheduled merchant-initiated transaction</p> <p>Z - unscheduled customer-initiated transaction</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent, the acceptable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p>

Variable Name	Type and Limits	Description
		if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
payment information <payment_information>	<i>String</i> 1-character numeric	Describes whether the transaction is the first or subsequent in the series Possible values: 0 - first transaction in a series (storing payment details provided by the cardholder) 2 - subsequent transactions (using previously stored payment details)

4 Vault Transaction Set

- 4.1 About the Vault Transaction Set
- 4.2 Vault Request DTD
- 4.3 Vault Response DTD
- 4.4 Vault Administrative Transactions
- 4.5 Vault Financial Transactions

4.1 About the Vault Transaction Set

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.

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

4.2 Vault Request DTD

```
<!-- The Resolver CA Request DTD -->

<!-- Main Elements -->

<!ELEMENT request (store_id, api_token, (res_add_cc | res_update_cc | res_delete | res_
lookup_full | res_lookup_masked | res_get_expiring | res_purchase_cc | res_preauth_cc | res_
ind_refund_cc | res_iscorporatecard))>
<!ELEMENT store_id (#PCDATA)>
<!ELEMENT api_token (#PCDATA)>
<!ELEMENT res_add_cc (cust_id?,phone?,email?,note?,pan?,expdate, crypt_type,avs_info?,cof_
info?)>
<!ELEMENT res_update_cc (data_key,cust_id?,phone?,email?,note?,pan?,expdate?,crypt_type?,avs_
info?,cof_info?)>
<!-- If "pan", "expdate", or "crypt_type" are sent then they are mandatory -->
<!ELEMENT res_delete (data_key)>
<!ELEMENT res_lookup_full (data_key)>
<!ELEMENT res_lookup_masked (data_key)>
<!ELEMENT res_get_expiring EMPTY> <!-- nothing else is required, returns all CC cards that
expire within the current or next month -->
<!ELEMENT res_purchase_cc (data_key,order_id,cust_id?,amount,crypt_type,cust_info?,avs_
info?,cvd_info?,recur?,cof_info?)>
<!-- if a cust_id is sent, it will be submitted with the purchase but not stored in profile -
->
<!-- if a cust_id is not sent, then will pull cust_id from profile and submit with purchase -
->
<!-- if no cust_id is sent or in profile then none will be sent with purchase -->
<!-- above cust_id behaviour also applies to avs_info -->
<!ELEMENT res_preauth_cc (data_key,order_id,cust_id?,amount,crypt_type,cust_info?,avs_
info?,cvd_info?,cof_info?)>
```

```
<!ELEMENT res_ind_refund_cc (data_key,order_id,cust_id?,amount,crypt_type)>
<!ELEMENT res_iscorporatecard (data_key)>
<!ELEMENT res_card_verification_cc (data_key,order_id,crypt_type,avs_info?,cvd_info?,cof_
info?)>
<!ELEMENT res_forcepost_cc (data_key,order_id,cust_id?,amount,crypt_type,auth_code)>
<!ELEMENT res_temp_add (pan, expdate, crypt_type, duration, data_key_format?)>
<!ELEMENT res_add_token (data_key,crypt_type,expdate,cof_info,cust_id?,avs_
info?,email?,phone?,note?,data_key_format?)>

<!ELEMENT data_key (#PCDATA)>
<!ELEMENT order_id (#PCDATA)>
<!ELEMENT cust_id (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT note (#PCDATA)>
<!ELEMENT pan (#PCDATA)>
<!ELEMENT expdate (#PCDATA)>
<!ELEMENT crypt_type (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT auth_code (#PCDATA)>

<!-- start AVS -->
<!ELEMENT avs_info (avs_street_number, avs_street_name, avs_zipcode)>
<!ELEMENT avs_street_number (#PCDATA)>
<!ELEMENT avs_street_name (#PCDATA)>
<!ELEMENT avs_zipcode (#PCDATA)>

<!-- start CVD -->
<!ELEMENT cvd_info (cvd_indicator, cvd_value)>
<!ELEMENT cvd_indicator (#PCDATA)>
<!ELEMENT cvd_value (#PCDATA)>

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

<!-- start COF -->
<!ELEMENT cof_info (payment_indicator, payment_information, issuer_id)>
<!ELEMENT payment_indicator (#PCDATA)>
<!ELEMENT payment_information (#PCDATA)>
<!ELEMENT issuer_id (#PCDATA)>

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

```

<!ELEMENT name (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
<!ELEMENT product_code (#PCDATA)>
<!ELEMENT extended_amount (#PCDATA)>
<!ELEMENT tax1 (#PCDATA)>
<!ELEMENT tax2 (#PCDATA)>
<!ELEMENT tax3 (#PCDATA)>
<!ELEMENT shipping_cost (#PCDATA)>

```

4.3 Vault Response DTD

```

<!-- The Response DTD -->

<!-- Main Elements -->

<!ELEMENT response receipt>
<!ELEMENT receipt (DataKey, ReceiptId, ReferenceNum, ResponseCode, AuthCode, ISO, Message,
TransTime, TransDate, TransType, Complete, TransAmount, CardType,
TransID, TimedOut, CvdResultCode, AvsResultCode, RecurSuccess,
CorporateCard, ResSuccess, PaymentType, ResolveData, IssuerId)>
<!ELEMENT DataKey (#PCDATA)>
<!ELEMENT ReceiptId (#PCDATA)>
<!ELEMENT ReferenceNum (#PCDATA)>
<!ELEMENT ResponseCode (#PCDATA)>
<!ELEMENT AuthCode (#PCDATA)>
<!ELEMENT ISO (#PCDATA)>
<!ELEMENT TransTime (#PCDATA)>
<!ELEMENT TransDate (#PCDATA)>
<!ELEMENT TransType (#PCDATA)>
<!ELEMENT Complete (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
<!ELEMENT TransAmount (#PCDATA)>
<!ELEMENT CardType (#PCDATA)>
<!ELEMENT TransID (#PCDATA)>
<!ELEMENT TimedOut (#PCDATA)>
<!ELEMENT CvdResultCode (ECR)>
<!ELEMENT AvsResultCode (#PCDATA)>
<!ELEMENT RecurSuccess (#PCDATA)>
<!ELEMENT CorporateCard (#PCDATA)> <!-- true or false -->
<!ELEMENT ResSuccess (#PCDATA)> <!-- true or false -->
<!ELEMENT PaymentType (#PCDATA)> <!-- cc -->
<!ELEMENT IssuerId (#PCDATA)>
<!ELEMENT ResolveData (data_key?, payment_type?, cust_id, phone, email, note,
masked_pan?, pan?, expdate?, crypt_type?,
avs_street_number?, avs_street_name?, avs_zipcode?)>

<!-- the following are only returned with res_get_expiring -->
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT payment_type (#PCDATA)>

<!-- the following are always returned, even if they are blank in the profile -->
<!ELEMENT cust_id (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT note (#PCDATA)>

<!-- the following are depending on what is stored in the profile -->
<!ELEMENT masked_pan (#PCDATA)>
<!ELEMENT expdate (#PCDATA)>
<!ELEMENT crypt_type (#PCDATA)>
<!ELEMENT avs_street_number (#PCDATA)>
<!ELEMENT avs_street_name (#PCDATA)>
<!ELEMENT avs_zipcode (#PCDATA)>

```

```
<!-- the following are only returned with res_lookup_full -->
<!ELEMENT pan (#PCDATA)>
```

4.4 Vault Administrative Transactions

Administrative transactions allow you to perform such tasks as creating new Vault profiles, deleting existing Vault profiles and updating profile information.

Some Vault Administrative Transactions require the Credential on File object to be sent with the **issuer ID** field only.

4.4.1 Vault Add Credit Card – res_add_cc

Creates a new credit card profile, and generates a unique data key which can be obtained from the Receipt object.

This data key is the profile identifier that all future financial Vault transactions will use to associate with the saved information.

XML transaction object

```
<res_add_cc>
```

Vault Add Credit Card transaction object definition

```
<!ELEMENT res_add_cc (cust_id?, phone?, email?, note?, pan, expdate, crypt_type, avs_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

Vault Add Credit Card transaction request fields – Required

Variable Name	Type and Limits	Description
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMY date format that is presented on the card.
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure)

Variable Name	Type and Limits	Description
		<p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Vault Add Credit Card transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
phone number <phone>	<p><i>String</i></p> <p>30-character alphanumeric</p>	<p>Customer's phone number</p> <p>Can be sent in when creating or updating a Vault profile</p>
email address <email>	<p><i>String</i></p> <p>30-character alphanumeric</p>	<p>Customer's email address</p> <p>Can be sent in when creating or updating a Vault profile</p>
note <note>	<p><i>String</i></p> <p>30-character alphanumeric</p>	<p>Used for any supplementary information related to the customer</p> <p>Can be sent in when creating or updating a Vault profile</p>
AVS Information	<i>Object</i>	Contains fields applying to the

Variable Name	Type and Limits	Description
<avs_info>	N/A	Address Verification Service (AVS) e-fraud tool

4.4.2 Vault Update Credit Card – res_update_cc

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

XML transaction object

```
<res_update_cc>
```

Vault Update Credit Card transaction object definition

```
<!ELEMENT res_update_cc (data_key, cust_id?, phone?, email?, note?, pan?, expdate?, crypt_type?, avs_info?, cof_info?)>
```

```
<!-- If "pan", "expdate", or "crypt_type" are sent then they are mandatory -->
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Vault Update Credit Card transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	String 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>

Vault Update Credit Card transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
phone number <phone>	<i>String</i> 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	<i>String</i> 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	<i>String</i> 30-character alphanumeric	Used for any supplementary information related to the customer Can be sent in when creating or updating a Vault profile
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits—field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. <div> NOTE: This is the reverse of the MMY date format that is presented on the card. </div>
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Instalment 4 – Mail Order / Telephone Order—Unknown

Variable Name	Type and Limits	Description
		<p>classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
<p>AVS Information</p> <p><avs_info></p> <div>For information on request fields for this object, see xrefHere</div>	<p>String</p> <p>N/A</p>	<p>Contains fields applying to the Address Verification Service (AVS) e-fraud tool</p>
<p>Credential on File Information</p> <p><cof_info></p> <div>For information on request fields for this object, see xrefHere</div>	<p>String</p> <p>N/A</p>	<p>Required when storing cardholder credentials or using these credentials in subsequent transactions.</p>

4.4.3 Vault Delete – res_delete

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

NOTE: Once a profile is deleted, the information that was saved within can no longer be retrieved.

XML transaction object

<res_delete>

Vault Delete transaction object definition

<!ELEMENT res_delete (data_key)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <p>NOTE: Only use once per transaction and within two minutes of the original</p>

Variable Name	Type and Limits	Description
		transaction request; if the status check request times out, do not send again, as additional investigation is required

Vault Delete transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered

4.4.4 Vault Lookup Full – res_lookup_full

Verifies what is currently saved under the Vault profile associated with the given data key. The response to this transaction returns the latest active data for that profile.

Unlike Vault Lookup Masked (which returns a masked credit card number), this transaction returns both the masked and unmasked credit card number.

XML transaction object

```
<res_lookup_full>
```

Vault Lookup Full transaction object definition

```
<!ELEMENT res_lookup_full (data_key)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your

Variable Name	Type and Limits	Description
		<p>test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
<p>status check</p> <p><status_check></p>	<p><i>Boolean</i></p> <p>true/false</p>	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Vault Lookup Full transaction request fields – Required

Variable Name	Type and Limits	Description
<p>data key</p> <p><data_key></p>	<p><i>String</i></p> <p>25-character alphanumeric</p>	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>

4.4.5 Vault Lookup Masked – res_lookup_masked

Verifies what is currently saved under the Vault profile associated with the given data key. The response to this transaction returns the latest active data for that profile.

Unlike Vault Lookup Full (which returns both the masked and the unmasked credit card numbers), this transaction only returns the masked credit card number.

XML transaction object

```
<res_lookup_masked>
```

Vault Lookup Masked transaction object definition

```
<!ELEMENT res_lookup_masked (data_key)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true

Variable Name	Type and Limits	Description
		NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

Vault Lookup Masked transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered

4.4.6 Vault Is Corporate Card – res_iscorporatecard

Determines whether a profile has a corporate card registered within it.

After sending the transaction, the response field to the Receipt object's `getCorporateCard` method is either true or false depending on whether the associated card is a corporate card.

XML transaction object

<res_iscorporatecard>

Vault Is Corporate Card transaction object definition

<!ELEMENT res_iscorporatecard (data_key)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your

Variable Name	Type and Limits	Description
		<p>test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
<p>status check</p> <p><status_check></p>	<p><i>Boolean</i></p> <p>true/false</p>	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Vault Is Corporate Card transaction request fields – Required

Variable Name	Type and Limits	Description
<p>data key</p> <p><data_key></p>	<p><i>String</i></p> <p>25-character alphanumeric</p>	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>

4.4.7 Vault Get Expiring – res_get_expiring

Verifies which profiles have credit cards that are expiring during the current and next calendar month.

EXAMPLE: if you are processing this transaction on September 30, then it will return all cards that expire(d) in September and October of this year.

When generating a list of profiles with expiring credit cards, only the masked credit card numbers are returned. Can be performed no more than 2 times on any given calendar day.

XML transaction object

<res_get_expiring>

Vault Get Expiring transaction object definition

```
<!ELEMENT res_get_expiring EMPTY> <!-- nothing else is required, returns all CC cards that expire within the current or next month -->
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter val-</p>

Variable Name	Type and Limits	Description
		<p>ues, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Vault Get Expiring transaction request fields – Required

Vault Get Expiring does not require any transaction request variables.

4.4.8 Vault Temporary Token Add – res_temp_add

Creates a new temporary token credit card profile. This transaction requires a duration to be set to indicate how long the temporary token is to be stored for. This is the API version of Hosted Tokenization.

During the lifetime of this temporary token, it may be used for any other vault transaction before it is permanently deleted from the system. Maximum duration is 15 minutes.

XML transaction object

```
<res_temp_add>
```

Vault Temporary Token Add transaction object definition

```
<!ELEMENT res_temp_add (pan, expdate, crypt_type, duration, data_key_format?)
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Variable Name	Type and Limits	Description
		moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

Vault Temporary Token Add transaction request fields – Required

Variable Name	Type and Limits	Description
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits—field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMY Y date format that is presented on the card.
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Installment

Variable Name	Type and Limits	Description
		<p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
<p>duration</p> <p><duration></p>	<p><i>String</i></p> <p>3-character numeric</p> <p>maximum 900 seconds</p>	<p>Amount of time the temporary token should be available</p>

Vault Temporary Token Add transaction request fields – Optional

Variable Name	Type and Limits	Description
<p>data key format</p> <p><data_key_format></p>	<p><i>String</i></p> <p>2-character alphanumeric</p>	<p>Specifies the data key format being returned</p> <p>If left blank, data key format will default to 25-character alphanumeric</p> <p>Possible values:</p> <p>0 – 25 character alphanumeric data key</p> <p>0U – unique 25-character alphanumeric data key</p>

4.4.9 Vault Add Token – res_add_token

Converts a Hosted Tokenization temporary token to a permanent Vault token.

A temporary token is valid for 15 minutes after it is created. This transaction must be performed within that time frame if the token is to be changed to a permanent one for future use.

Vault Add Token – res_add_token transaction object definition

```
<!ELEMENT res_add_token (data_key, crypt_type, expdate, cof_info, cust_id?,  
avs_info?, email?, phone?, note?, data_key_format?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check</p> </div>

Variable Name	Type and Limits	Description
		request times out, do not send again, as additional investigation is required

Vault Add Token – res_add_token transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Installment 4 – Mail Order / Telephone Order—Unknown classification 5 – Authenticated e-commerce transaction (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator , as follows: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7

Variable Name	Type and Limits	Description
		<p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYYY	Expiry date of the credit card, in YYYY format.
NOTE: This field is optional if you already collected it using the the Hosted Tokenization solution; otherwise, it is required		NOTE: This is the reverse of the MMY Y date format that is presented on the card.

Vault Add Token – res_add_token transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
	NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	
phone number <phone>	<i>String</i> 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	<i>String</i> 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	<i>String</i> 30-character alphanumeric	Used for any supplementary information related to the customer Can be sent in when creating or updating a Vault profile
AVS Information	<i>Object</i>	Contains fields applying to the Address Verification Service (AVS) e-

Variable Name	Type and Limits	Description
<avs_info>	N/A	fraud tool For more information about AVS, see 10.1.2 AVS Information Object
data key format <data_key_format>	<i>String</i> 2-character alphanumeric	Specifies the data key format being returned If left blank, data key format will default to 25-character alphanumeric Possible values: 0 – 25 character alphanumeric data key 0U – unique 25-character alphanumeric data key

4.5 Vault Financial Transactions

After a financial transaction is complete, the response fields indicate all the values that are currently saved under the profile that was used.

4.5.1 Customer ID Changes

Some financial transactions take the customer ID as an optional value. The customer ID may or may not already be in the Vault profile when the transaction is sent. Therefore, it is possible to change the value of the customer ID by performing a financial transaction

The table below shows what the customer ID will be in the response field after a financial transaction is performed.

Table 1: Customer ID use in response fields

Already in profile?	Passed in?	Version used in response
No	No	Customer ID not used in transaction
No	Yes	Passed in
Yes	No	Profile
Yes	Yes	Passed in

4.5.2 Purchase with Vault – res_purchase_cc

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

The data key may be a temporary one generated used Hosted Tokenization, or may be a permanent one from the Vault.

XML transaction object

```
<res_purchase_cc>
```

Purchase with Vault transaction object definition

```
<!ELEMENT res_purchase_cc (data_key,order_id,cust_id?,amount,crypt_type,cust_info?,avs_info?,cvd_info?,recur?,cof_info?)>
```

```
<!-- if a cust_id is sent, it will be submitted with the purchase but not stored in profile -->
```

```
<!-- if a cust_id is not sent, then will pull cust_id from profile and submit with purchase -->
```

```
<!-- if no cust_id is sent or in profile then none will be sent with purchase -->
```

```
<!-- above cust_id behaviour also applies to avs_info -->
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Purchase with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
order ID <order_id>	<i>String</i> 50-character alphanumeric-A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>

Variable Name	Type and Limits	Description
	<div>EXAMPLE: 1234567.89</div>	
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7

Purchase with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div>NOTE: Some special characters are not allowed:</div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>

Variable Name	Type and Limits	Description
	< > \$ % = ? ^ { } [] \	
Customer Information <cust_info>	Object N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
For information on request fields for this object, see xrefHere		
AVS Information <avs_info>	Object N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
For information on request fields for this object, see xrefHere		
CVD Information <cvd_info>	Object N/A	Contains fields related to the Card Validation Digits e-fraud tool
For information on request fields for this object, see xrefHere		
Credential on File Information <cof_info>	Object N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
For information on request fields for this object, see xrefHere		

4.5.3 Pre-Authorization with Vault – res_preauth_cc

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

The data key may be a temporary one generated used Hosted Tokenization, or may be a permanent one from the Vault.

XML transaction object

<res_preauth_cc>

Pre-Authorization with Vault transaction object definition

```
<!ELEMENT res_preauth_cc (data_key, order_id, cust_id?, amount, crypt_type,
cust_info?, avs_info?, cvd_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

s	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Pre-Authorization with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key	String	Unique identifier for a Vault profile,

Variable Name	Type and Limits	Description
<data_key>	25-character alphanumeric	and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	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>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant

Variable Name	Type and Limits	Description
		<p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Pre-Authorization with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
Customer Information <cust_info> <div>For information on request fields for this object, see xrefHere</div>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> <div>For information on request fields for this object, see xrefHere</div>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information	<i>Object</i>	Contains fields related to the Card Validation Digits e-fraud tool

Variable Name	Type and Limits	Description
<cvd_info>	N/A	
<div>For information on request fields for this object, see xrefHere</div>		
Credential on File Information <cof_info>	Object N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
<div>For information on request fields for this object, see xrefHere</div>		

4.5.4 Independent Refund with Vault – res_ind_refund_cc

This transaction uses the data key to identify a previously registered credit card profile in Vault. The details saved within the profile are then submitted to perform an Independent Refund transaction.

XML transaction object

<res_ind_refund_cc>

Independent Refund with Vault transaction object definition

```
<!ELEMENT res_ind_refund_cc (data_key, order_id, cust_id?, amount, crypt_type)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p>

Variable Name	Type and Limits	Description
---------------	-----------------	-------------

Production: <https://www3.moneris.com/mpg/>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

Independent Refund with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - . : @ spaces	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	<i>String</i>	Transaction dollar amount

Variable Name	Type and Limits	Description
<amount>	<p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p> <p>EXAMPLE: 1234567.89</p>	<p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
<p>electronic commerce indicator</p> <p><crypt_type></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <p>1 – Mail Order / Telephone Order—Single</p> <p>2 – Mail Order / Telephone Order—Recurring</p> <p>3 – Mail Order / Telephone Order—Instalment</p> <p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Independent Refund with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center

4.5.5 Force Post with Vault – res_forcepost_cc

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

XML transaction object

<res_forcepost_cc>

Force Post with Vault transaction object definition

```
<!ELEMENT res_forcepost_cc (data_key, order_id, cust_id?, amount, crypt_type, auth_code)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Force Post with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
order ID <order_id>	<i>String</i> 50-character alphanumeric-A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>

Variable Name	Type and Limits	Description
	<div>EXAMPLE: 1234567.89</div>	
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
authorization code <auth_code>	<i>String</i> 8-character alphanumeric	An authorization code required to carry out a Force Post; provided in the transaction response from the issuing bank

Force Post with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID	<i>String</i>	Merchant-defined field that can be

Variable Name	Type and Limits	Description
<cust_id>	50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	used as an identifier Searchable from the Moneris Merchant Resource Center

4.5.6 Card Verification with Vault – res_card_verification_cc

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

The data key may be a temporary one generated used Hosted Tokenization, or may be a permanent one from the Vault.

XML transaction object

<res_card_verification_cc>

Card Verification with Vault transaction object definition

```
<!ELEMENT res_card_verification_cc (data_key, order_id, crypt_type, avs_info?, cvd_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Card Verification with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
order ID <order_id>	<i>String</i> 50-character alphanumeric-A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <p>1 – Mail Order / Telephone Order—Single</p> <p>2 – Mail Order / Telephone Order—Recurring</p>

Variable Name	Type and Limits	Description
		<p>3 – Mail Order / Telephone Order—Instalment</p> <p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Card Verification with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
AVS Information <avs_info>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
<div>For information on request fields for this object, see xrefHere</div>		
CVD Information <cvd_info>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
<div>For information on request fields for this object, see xrefHere</div>		
Credential on	<i>Object</i>	Required when storing cardholder cre-

Variable Name	Type and Limits	Description
File Information <cof_info> <div>For information on request fields for this object, see xrefHere</div>	N/A	dentials or using these credentials in subsequent transactions.

5 3-D Secure 2.0 Transactions

- 5.1 About 3-D Secure 2.0
- 5.2 Building Your 3-D Secure 2.0 Integration
- 5.3 Implementing Card Lookup Request
- 5.4 Implementing MPI 3DS Authentication Request
- 5.5 Handling the Challenge Flow
- 5.6 Performing the Authorization
- 5.7 Testing Your 3-D Secure 2.0 Integration
- 5.8 Moving to Production With 3-D Secure 2.0
- 5.9 3-D Secure 2.0 TransStatus Codes
- 5.10 CAVV Result Codes

5.1 About 3-D Secure 2.0

3-D Secure 2.0 is an EMVCo payment authentication protocol designed to reduce card not present fraud by making a risk assessment based on transaction and device data, while also supporting further risk minimization measures, such as a challenge to the cardholder. In some cases, a liability shift takes effect for certain card-not-present fraud-related chargebacks enabling the merchant to provide goods and services with confidence.

The Moneris Gateway can enable transactions using the 3-D Secure protocol via Moneris 3DS Server and Access Control Server (ACS).

Moneris Gateway supports the following 3-D Secure implementations:

- Visa Secure
- Mastercard Identity Check
- American Express SafeKey (please note: American Express only supports authentication requests for merchants who have an Amex OFI merchant account)

5.1.1 3-D Secure Implementations

Visa Secure, Mastercard Identity Check and American Express SafeKey are programs based on the 3-D Secure Protocol to improve the security of online transactions.

These programs involve authentication of the cardholder during an online e-commerce transaction.

Authentication is based on the issuer's selected method of authentication.

The following are examples of authentication methods:

- Risk-based authentication
- Dynamic passwords
- Static passwords

Some benefits of these programs are reduced risk of fraudulent transactions and protection against chargebacks for certain fraudulent transactions.

5.1.2 Out of Scope/Not Supported

- In-app
- 3RI

5.1.3 Version Compatibility

All development to the Moneris API must be able to support the addition of new fields in the response and new error conditions in the response. Otherwise any changes that affect backwards compatibility will be communicated by Moneris Solutions with an appropriate period of notice. When developing to the solution it is recommended to validate for success state of the request and then handle errors states separately and ensure there is a final catch for any unexpected/undocumented errors that are returned.

5.1.4 Upgrading from 3-D Secure 1.0 to 3-D Secure 2.0

The 3DS 2.0 API is different from the 3DS 1.0 API therefore developers will have to complete the steps described in the section 5.2 Building Your 3-D Secure 2.0 Integration.

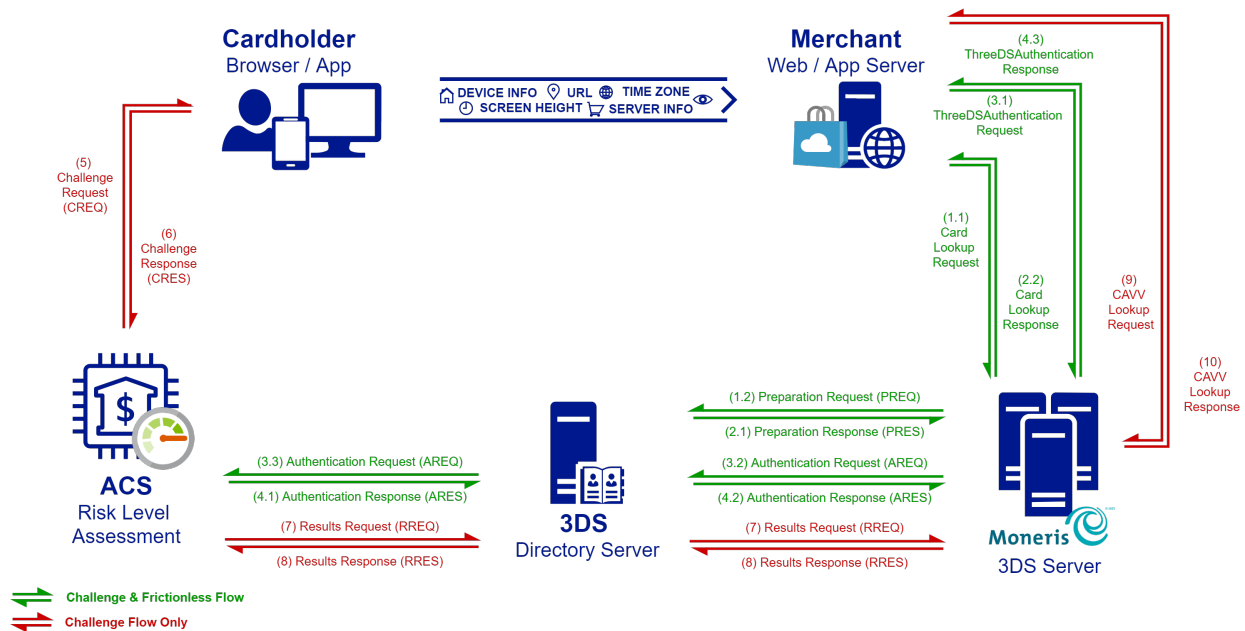
5.2 Building Your 3-D Secure 2.0 Integration

- 5.2.1 Activating 3-D Secure Functionality
- 5.2.2 Transaction Flow for 3-D Secure 2.0

5.2.1 Activating 3-D Secure Functionality

To activate Visa Secure, Mastercard Identity Check and/or American Express SafeKey transaction functionality, call Moneris Sales Support at 1-855-465-4980 to have Moneris enroll you in the program(s) and enable the functionality on your account.

5.2.2 Transaction Flow for 3-D Secure 2.0



The 3DS 2.0 API is called when the customer wishes to checkout. An optional card lookup request can be performed to initiate cardholder browser fingerprinting. Once the fingerprint is complete, or as a first step if not performing a fingerprint, the transactional information can then be transmitted to the 3DS 2.0 service so a risk assessment may be initiated.

The flow can then proceed in one of two ways. The two different flows are referred to as “frictionless” and “challenge”.

The “frictionless” flow is transparent to a cardholder. If the issuing financial institution has enough information to make a risk assessment and assume liability, this will manifest itself as with an authentication attempt or success with an accompanying CAVV value. No cardholder challenge is presented.

In the “challenge” flow the issuing financial institution may wish to take a further step and issue a challenge to the cardholder. In this case the cardholder’s browser gets re-directed to the issuer’s 3DS platform for authentication. Once this challenge is complete, the cardholder browser is again re-directed back to the merchant’s site. The merchant’s server then issues a server-to-server request in order to obtain the CAVV value from Moneris.

5.2.3 Mpi2Request Object and 3-D Secure Authentication

The authentication transactions for 3-D Secure 2.0 transactions are represented as part of the **Mpi2Request** object.

There are three transactions that are sent under Mpi2Request:

- Card Lookup Request
- MPI 3DS Authentication Request
- Cavv Lookup Request

Once authentication is complete, Purchase with 3-D Secure – cavv_purchase or Pre-Authorization with 3-D Secure – cavv_preauth are performed subsequently for authorization.

5.2.4 3-D Secure 2.0 Request DTD

```
<!ELEMENT Mpi2Request (store_id, api_token, (card_lookup | threads_authentication | cavv_
lookup))>
<!ELEMENT card_lookup (order_id, (data_key | pan), notification_url)>
<!ELEMENT threads_authentication (order_id, cardholder_name, (data_key | pan), expdate,
amount, currency?, threads_completion_ind, request_type, notification_url, purchase_date,
challenge_window_size, bill_address1?, bill_province?, bill_city?, bill_postal_code?, bill_
country?, ship_address1?, ship_province?, ship_city?, ship_postal_code?, ship_country?,
browser_useragent, browser_java_enabled, browser_screen_height, browser_screen_width,
browser_language, email?, request_challenge?)>
<!ELEMENT cavv_lookup (cres)>

<!-- start 3DS 2.0 specific fields -->
<!ELEMENT threads_version (#PCDATA)>
<!ELEMENT threads_server_trans_id (#PCDATA)>
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT notification_url (#PCDATA)>
<!ELEMENT cardholder_name (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT threads_completion_ind (#PCDATA)>
<!ELEMENT request_type (#PCDATA)>
<!ELEMENT purchase_date (#PCDATA)>
<!ELEMENT challenge_window_size (#PCDATA)>
<!ELEMENT bill_address1 (#PCDATA)>
<!ELEMENT bill_province (#PCDATA)>
<!ELEMENT bill_city (#PCDATA)>
<!ELEMENT bill_postal_code (#PCDATA)>
<!ELEMENT bill_country (#PCDATA)>
<!ELEMENT ship_address1 (#PCDATA)>
<!ELEMENT ship_province (#PCDATA)>
<!ELEMENT ship_city (#PCDATA)>
<!ELEMENT ship_postal_code (#PCDATA)>
<!ELEMENT ship_country (#PCDATA)>
<!ELEMENT browser_useragent (#PCDATA)>
<!ELEMENT browser_java_enabled (#PCDATA)>
<!ELEMENT browser_screen_height (#PCDATA)>
<!ELEMENT browser_screen_width (#PCDATA)>
```

```
<!--ELEMENT browser_language (#PCDATA)>
<!--ELEMENT request_challenge (#PCDATA)>
<!--ELEMENT cres (#PCDATA)>
```

5.2.5 3-D Secure 2.0 Response DTD

```
<!-- The following are only applicable to 3DS 2.0 transactions -->
<!--ELEMENT MessageType (#PCDATA)>
<!--ELEMENT ThreeDSMethodURL (#PCDATA)>
<!--ELEMENT ThreeDSMethodData (#PCDATA)>
<!--ELEMENT ChallengeURL (#PCDATA)>
<!--ELEMENT ChallengeData (#PCDATA)>
<!--ELEMENT ChallengeCompletionIndicator (#PCDATA)>
<!--ELEMENT TransStatus (#PCDATA)>
<!--ELEMENT ThreeDSSTransId (#PCDATA)>
<!--ELEMENT ECI (#PCDATA)>
<!--ELEMENT Cavv (#PCDATA)>
```

5.3 Implementing Card Lookup Request

The CardLookup request verifies the applicability of 3DS 2.0 on the card and returns the 3DS Method URL. That is used for device fingerprinting. This request is optional, it may increase the chance of a frictionless flow.

The threeDSMethodURL & threeDSMethodData are returned to the merchant server on the CardLookup response. The threeDSMethodData can be transmitted to the threeDSMethodURL via a browser post in order to supplement the authentication request with device data pertaining to the cardholder's browser.

The threeDSMethodData must be sent via HTTP POST to the threeDSMethodURL in a hidden iFrame.

In your implementation, use the following URLs as Host, depending on the development stage:

Testing:

esqa.moneris.com

Production:

www3.moneris.com

5.3.1 Card Lookup Request

XML transaction object

```
<card_lookup>
```

TransactionTopicName transaction object definition

```
<!--ELEMENT card_lookup (order_id, (data_key | pan), notification_url)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

TransactionTopicName transaction request fields – Required

Variable Name	Type and Limits	Description
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere	String	

Variable Name	Type and Limits	Description
<xmlvariableHere>		
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	

TransactionTopicName transaction request fields – Optional

Variable Name	Type and Limits	Description
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	
humanSnippetHere <xmlvariableHere>	String	

5.3.2 Card Lookup Request – mpiCardLookup

Card Lookup Request transaction object definition

```
MpiCardLookup mpiCardLookup = new MpiCardLookup();
```

```
$mpiCardLookup = new MpiCardLookup();
```

```
MpiCardLookup mpiCardLookup = new MpiCardLookup();
```


HttpPostRequest object for Card Lookup Request transaction

```

HttpPostRequest mpgReq = new HttpPostRequest();

mpgReq.setTransaction(mpiCardLookup);

$mpgRequest = new mpgRequest($mpgTxn);

$mpgTxn = new mpgTransaction($mpiCardLookup);

HttpPostRequest mpgReq = new HttpPostRequest();

mpgReq.SetTransaction(mpiCardLookup);

```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Card Lookup Request transaction request fields – Required

NOTE: Either a pan or a data_key must be passed in the request

Variable Name	Type and Limits	Set Method
order ID	String 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	mpiCardLookup

Variable Name	Type and Limits	Set Method
credit card number	<i>String</i> max 20-character alpha-numeric	mpiCardLookup
data key	<i>String</i> 25-character alphanumeric	mpiCardLookup.setData(data_key); SetData(data_key);
notification URL	<i>String</i> 256-character alpha-numeric	mpiCardLookup \$mpiCardLookup

Sample Card Lookup Request

5.4 Implementing MPI 3DS Authentication Request

The MPI 3DS Authentication Request is used to start the validation process of the card. The result of this request determines whether 3DS 2.0 is supported by the card and what type of authentication is required.

In your implementation, use the following URLs as Host, depending on the development stage:

Testing:

esqa.moneris.com

Production:

www3.moneris.com

5.4.1 MPI 3DS Authentication Request

XML transaction object

```
<threads_authentication>
```

MPI 3DS Authentication Request transaction object definition

```
<!ELEMENT threads_authentication (order_id, cardholder_name, (data_key | pan),
expdate, amount, currency?, threads_completion_ind, request_type,
notification_url, purchase_date, challenge_window_size, bill_address1?, bill_
province?, bill_city?, bill_postal_code?, bill_country?, ship_address1?, ship_
province?, ship_city?, ship_postal_code?, ship_country?, browser_useragent,
browser_java_enabled, browser_screen_height, browser_screen_width, browser_
language, email?, request_challenge?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

MPI 3DS Authentication Request 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
	a-Z A-Z 0-9 _ - : . @ spaces	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.
cardholder name <cardholder_name>	String 45-character alphanumeric	Name of the cardholder
data key <data_key> OR credit card number <pan>	String data key limits: 25-character alphanumeric credit card number limits: max 20-character alphanumeric	data key description: Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered credit card number description: Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	String 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMY date format that is presented on the card.
amount <amount>	String 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point EXAMPLE: 1234567.89	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99

Variable Name	Type and Limits	Description
3DS completion indicator <threeds_completion_ind>	<i>String</i> 1-character alphanumeric	indicates whether 3ds method MpiCardLookup was successfully completed Allowable values: Y = Successfully completed N = Did not successfully complete U = Unavailable
request type <request_type>	<i>String</i> 2-character alphanumeric	Allowable values: 01 = cardholder initiated payment 02 = recur
notification URL <notification_url>	<i>String</i> 256-character alphanumeric	Website URL that will receive 3DS Method Completion response from ACS
purchase date <purchase_date>	<i>String</i> 14-character numeric YYYYMMDDHHMMSS	Numerical representation of the date (year, month, day) and time (hours, minutes, seconds) of purchase
challenge window size <challenge_window_size>	<i>String</i> 2-character alphanumeric	Relates to the rendering of the ACS challenge within the browser. Allowable values: 01 = 250 x 400 02 = 390 x 400 03 = 500 x 600 04 = 600 x 400 05 = Full screen
browser user agent <browser_useragent>	<i>String</i> 2048-character alphanumeric	Browser User Agent
browser java enabled <browser_java_enabled>	<i>String</i> 1-character alphabetic	Indicates whether Java is enabled in the browser Allowable values:

Variable Name	Type and Limits	Description
		T = True F = False
browser screen height <browser_screen_height>	String 6-character numeric	Pixel height of cardholder screen
browser screen width <browser_screen_width>	String 6-character numeric	Pixel width of cardholder screen
browser language <browser_language>	String 8-character alphanumeric	As defined in IETF BCP47
email <email>	String 254-character alpha-numeric	
request challenge <request_challenge>	String 1-character alphabetic	Indicates whether a challenge is requested for this transaction Allowable values: Y = Yes N = No

MPI 3DS Authentication Request transaction request fields – Optional

Variable Name	Type and Limits	Description
currency <currency>	String 3-character numeric	ISO 4217 3 digit currency code CAD = 124 USD = 840
<div> NOTE: This field should not be sent unless Multi Currency Pricing is enabled on your merchant account </div>		
billing address <bill_address1>	String 50-character alphanumeric	Cardholder billing address
billing province	String	Defined in country subdivision ISO

Variable Name	Type and Limits	Description
<bill_province>	3-character alphanumeric Defined in country sub-division ISO 3166-2	3166-2
billing city <bill_city>	String 50-character alphanumeric	Cardholder billing city
billing postal code <bill_postal_code>	String 16-character alphanumeric	Cardholder billing postal code
billing country <bill_country>	String 3-character numeric Defined as 3 digit country code ISO 3166-1	Defined as 3 digit country code ISO 3166-1
shipping address <ship_address1>	String 50-character alphanumeric	Shipping destination address
shipping province <ship_province>	String 3-character alphanumeric defined in country sub-division ISO 3166-2	Shipping destination province Defined in country subdivision ISO 3166-2
shipping city <ship_city>	String 50-character alphanumeric	Shipping destination city
shipping postal code <ship_postal_code>	String 16-character alphanumeric	Shipping destination postal code
ship country <ship_country>	String 3-character numeric defined as 3 digit country code ISO 3166-1	Shipping destination country Defined as 3-digit country code in ISO 3166-1
email <email>	String 254-character alpha-numeric	Cardholder email address

Variable Name	Type and Limits	Description
request challenge <request_challenge>	String 1-character alphabetic	Indicates whether a challenge is requested for this transaction Allowable values: Y = Yes N = No
humanSnippetHere <xmlvariableHere>	String	

5.5 Handling the Challenge Flow

If you get a TransStatus = “C” in your threeDSAuthentication Response, then a form must be built and POSTed to the URL provided.

The form can be dynamically generated and added to the DOM and submitted or created and submitted in a manner that suits your environment. This can be built as a full page redirect or presented as an inline iframe or as a lightbox.

If you wish for this to be loaded inside a defined space it must conform to the size specified in the challengeWindowSize from the request. The “action” is retrieved from the ChallengeURL and the “creq” field is retrieved from the ChallengeData.

Below is a sample of a basic static form to help visualize the data and fields that need to be submitted.

```
<form method="POST" action="https://3dsurl.example.com/do3DS">
<input name="creq" value="thisissamplechallengedata1234567890">
</form>
```

5.5.1 Cavv Lookup Request

(Challenge Flow Only)

In the challenge flow, the 3DS server will POST a **cres** value back to the notificationURL provided in the threeDSAuthentication request once the cardholder has completed the challenge. The “cres” is then posted to the Moneris 3DS server in the CavvLookup request, the response to this request will include the result of the challenge, which will include the eci and the cavv if the challenge was successful.

XML transaction object

```
<cavv_lookup>
```

Cavv Lookup Request transaction object definition

```
<!ELEMENT cavv_lookup (cres)>
```


Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Cavv Lookup Request transaction request fields – Required

Variable Name	Type and Limits	Description
cres <cres>	String 200-character alphanumeric	Response data from the challenge

5.6 Performing the Authorization

Once the authentication is complete and a CAVV and ECI value are retrieved, these values can be sent to Moneris using the transactions Purchase with 3-D Secure – cavvPurchase or Pre-Authorization with 3-D Secure – cavvPreauth.

5.6.1 Purchase with 3-D Secure – cavv_purchase

The Purchase with 3-D Secure transaction follows a 3-D Secure MPI authentication. After receiving confirmation from the MPI ACS transaction, Purchase with 3-D Secure verifies funds on the customer's card, removes the funds and prepares them for deposit into the merchant's account.

XML transaction object

```
<cavv_purchase>
```

Purchase with 3-D Secure transaction object definition

```
<!ELEMENT cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv,
dynamic_descriptor?, cust_info?, avs_info?, cvd_info?, recur?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	Checks whether a previously sent transaction was processed successfully

Variable Name	Type and Limits	Description
		<p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

Purchase with 3-D Secure transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alpha-numeric-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
credit card number <pan>	<i>String</i> max 20-character alpha-numeric	<p>Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.</p>
expiry date <expdate>	<i>String</i> 4-character alphanumeric	<p>Expiry date of the credit card, in YYYYMM format.</p>

Variable Name	Type and Limits	Description
	YYMM	NOTE: This is the reverse of the MMYM date format that is presented on the card.
Cardholder Authentication Verification Value (CAVV) <cavv>	<i>String</i> 50-character alphanumeric	Value provided by the Moneris MPI or by a third-party MPI Sent in all 3-D Secure transactions, including Verified By Visa, MasterCard SecureCode, American Express SafeKey For Purchase and Pre-Authorization transactions with 3-D Secure for Apple Pay and Google Pay, the CAVV field contains the decrypted cryptogram

Purchase with 3-D Secure transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters

Variable Name	Type and Limits	Description
Customer Information <cust_info>	Object N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info>	Object N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info>	Object N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur>	Object N/A	Contains fields related to Recurring Billing
Credential on File Information <cof_info>	Object N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

5.6.2 Pre-Authorization with 3-D Secure – cavv_preauth

The Pre-Authorization with 3-D Secure transaction follows a 3-D Secure MPI authentication. After receiving confirmation from the MPI ACS transaction, the Pre-Authorization with 3-D Secure verifies funds on the customer's card, removes the funds and prepares them for deposit into the merchant's account.

XML transaction object

<cavv_preauth>

Pre-Authorization with 3-D Secure transaction object definition

```
<!ELEMENT cavv_preauth (order_id, cust_id? amount, pan, expdate, cavv,
dynamic_descriptor?, cust_info?, avs_info?, cvd_info?, cof_info?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation

Variable Name	Type and Limits	Description
		<p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Pre-Authorization with 3-D Secure transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alpha-numeric-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two</p>

Variable Name	Type and Limits	Description
	Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point EXAMPLE: 1234567.89	of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMY date format that is presented on the card.
Cardholder Authentication Verification Value (CAVV) <cavv>	<i>String</i> 50-character alphanumeric	Value provided by the Moneris MPI or by a third-party MPI Sent in all 3-D Secure transactions, including Verified By Visa, MasterCard SecureCode, American Express SafeKey For Purchase and Pre-Authorization transactions with 3-D Secure for Apple Pay and Google Pay, the CAVV field contains the decrypted cryptogram

Pre-Authorization with 3-D Secure transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor	<i>String</i>	Merchant-defined description sent on

Variable Name	Type and Limits	Description
<dynamic_descriptor>	20-character alphanumeric total of 22 characters including your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div>
Customer Information <cust_info>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur>	<i>Object</i> N/A	Contains fields related to Recurring Billing
Credential on File Information <cof_info>	<i>Object</i> N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

5.7 Testing Your 3-D Secure 2.0 Integration

In the testing stage of development:

1. Use the testing URL as Host for your requests:
esqa.moneris.com

2. In all Card Lookup Request transactions, make sure that you are using the testing version of your credentials for store ID and API token
3. In all MPI 3DS Authentication Request transactions, make sure that you are using the testing version of your credentials for store ID and API token
4. In all Cavv Lookup Request transactions, make sure that you are using the testing version of your credentials for store ID and API token

5.8 Moving to Production With 3-D Secure 2.0

Once you have finished testing your 3D Secure 2.0 integration, do the following to move the integration into production:

1. Use the production URL as Host for your requests:
`www3.moneris.com`
2. In all Card Lookup Request transactions, make sure that you are using the production version of your credentials for store ID and API token
3. In all MPI 3DS Authentication Request transactions, make sure that you are using the production version of your credentials for store ID and API token
4. In all Cavv Lookup Request transactions, make sure that you are using the production version of your credentials for store ID and API token

5.9 3-D Secure 2.0 TransStatus Codes

Value	Description	Comments
Y	Authenticated	Cardholder has been fully authenticated
A	Authentication Attempt	A proof of authentication attempt was generated
C	Challenge Required	Cardholder requires a challenge to complete authentication
U	Not Authenticated	Authentication could not be performed due to technical or other issue
N	Not Authenticated	Not authenticated

Value	Description	Comments
R	Not Authenticated	Not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted

5.10 CAVV Result Codes

The Cardholder Authentication Verification Value (CAVV), the Accountholder Authentication Value (AAV), and the American Express Verification Value (AEVV), are the values that allows Visa, Mastercard and American Express to validate the integrity of the Visa Secure, Mastercard Identity Check and American Express SafeKey transaction data. These values are passed back from the issuer to the merchant after the authentication has taken place. The merchant then integrates the CAVV/AAV/AEVV value into the authorization request using the Purchase or Pre-Authorization with 3-D Secure transaction type.

To summarize this process:

1. Merchant conducts 3-D Secure authentication request and receives CAVV/AAV/AEVV value in response
2. Merchant sends the CAVV/AAV/AEVV value to Moneris using the Purchase or Pre-Authorization with 3-D Secure transaction type and receives the CAVV result code in the response

The following tables describe the contents of the CAVV data response and what it means to the merchant.

5.10.1 Visa CAVV Result Codes

Visa CAVV result codes

Result Code	Message	Significance to Merchants
Blank	CAVV not present or not verified	Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks
0	CAVV authentication results invalid	Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks
1	CAVV failed validation (authentication)	Provided that you have implemented the Visa Secure process correctly, the liability for this transaction should remain with the Issuer for chargeback reason codes covered by Visa Secure.
2	CAVV passed validation	Fully authenticated transaction. There is a liab-

Result Code	Message	Significance to Merchants
	(authentication)	liability shift and the merchant is protected from chargebacks.
3, 8, A	CAVV passed validation (attempt)	Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks.
4, 7, 9	CAVV failed validation (attempt)	Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks.
6	CAVV not validated - Issuer not participating	Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks.
B	CAVV passed validation; information only	Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks
C	CAVV was not validated (attempt)	Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks.
D	CAVV was not validated (authentication)	Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks.

5.10.2 Mastercard CAVV Result Codes

Mastercard CAVV result codes

Result Code	Message	Significance to Merchants
0	Authentication failed	Not a Mastercard Identity Check transaction. No liability shift and merchant is not protected from chargebacks
1	Authentication attempted	Mastercard Identity Check has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks (international commercial cards excluded).
2	Authentication successful	Fully authenticated transaction. There is a liability shift and the merchant is protected from

Result Code	Message	Significance to Merchants
		chargebacks.

5.10.3 American Express CAVV Result Codes

American Express CAVV result codes

NOTE: American Express SafeKey is only available to American Express direct acquired merchants (i.e., not OptBlue merchants). Any questions pertaining to chargebacks, liability and disputes should be addressed to your American Express representative given that American Express is the acquirer of record for these merchants.

Result Code	Description
1	AEVV Failed - Authentication, Issuer Key
2	AEVV Passed - Authentication, Issuer Key
3	AEVV Passed - Attempt, Issuer Key
4	AEVV Failed - Attempt, Issuer Key
7	AEVV Failed - Attempt, Issuer not participating, Network Key
8	AEVV Passed - Attempt, Issuer not participating, Network Key
9	AEVV Failed - Attempt, Participating, Access Control Server (ACS) not available, Network Key
A	AEVV Passed - Attempt, Participating, Access Control Server (ACS) not available, Network Key
U	AEVV Unchecked

6 Multi-Currency Pricing (MCP)

- 6.1 About Multi-Currency Pricing (MCP)
- 6.2 Methods of Processing MCP Transactions
- 6.3 Multi-Currency Pricing (MCP) Request DTD
- 6.4 Multi-Currency Pricing (MCP) Response DTD
- 6.5 MCP Purchase
- 6.6 MCP Pre-Authorization
- 6.7 MCP Pre-Authorization Completion
- 6.8 MCP Purchase Correction
- 6.9 MCP Refund
- 6.10 MCP Independent Refund
- 6.11 MCP Purchase With Vault
- 6.12 MCP Pre-Authorization With Vault
- 6.13 MCP Independent Refund with Vault
- 6.14 MCP Get Rate
- 6.15 MCP Currency Codes
- 6.16 MCP Error Codes

6.1 About Multi-Currency Pricing (MCP)

Multi-currency pricing (MCP) is a financial service which allows businesses to price goods and services in a variety of foreign currencies, while continuing to receive settlement and reporting in Canadian dollars. MCP allows cardholders to shop, view prices and pay in the currency of their choice.

MCP is only available when processing Visa and Mastercard transactions.

NOTE: Use MCP only when processing transactions that involve foreign currency exchange; for transactions strictly in Canadian dollars, use the basic financial transaction requests

6.2 Methods of Processing MCP Transactions

There are two methods of processing multi-currency pricing transactions via the Moneris Gateway:

1. **Using the MCP Get Rate transaction** – this method is used to obtain a foreign exchange rate and locks that specific rate in for a limited time, and is applied in a subsequent transaction
2. **Without using MCP Get Rate** – this method sends a MCP transaction without performing the Get Rate request, and the foreign exchange rate is obtained at processing time

6.3 Multi-Currency Pricing (MCP) Request DTD

```
<!ELEMENT mcp_completion (order_id, txn_number, crypt_type, cust_id, dynamic_descriptor?,
ship_indicator?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?) >
<!ELEMENT mcp_ind_refund (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_preauth (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
<!ELEMENT mcp_purchase (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?,
wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
<!ELEMENT mcp_purchasecorrection (order_id, txn_number, crypt_type, cust_id)>
<!ELEMENT mcp_refund (order_id, amount, txn_number, crypt_type, cust_id, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_ind_refund_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_preauth_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_res_purchase_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_get_rate (mcp_version, rate_txn_type, rate_info)>
```

6.4 Multi-Currency Pricing (MCP) Response DTD

```
<!ELEMENT Rate (CardholderCurrencyCode, CardholderAmount, MerchantSettlementCurrency,
MerchantSettlementAmount, MCPRate, MCPErrorStatusCode, MCPErrorMessage)>
<!ELEMENT MCPRateToken (#PCDATA)>
<!ELEMENT RateTxnType (#PCDATA)>
<!ELEMENT RateInqStartTime (#PCDATA)>
<!ELEMENT RateInqEndTime (#PCDATA)>
<!ELEMENT RateValidityStartTime (#PCDATA)>
<!ELEMENT RateValidityEndTime (#PCDATA)>
<!ELEMENT RateValidityPeriod (#PCDATA)>
<!ELEMENT CardholderCurrencyCode (#PCDATA)>
<!ELEMENT CardholderAmount (#PCDATA)>
<!ELEMENT MerchantSettlementCurrency (#PCDATA)>
<!ELEMENT MerchantSettlementAmount (#PCDATA)>
<!ELEMENT MCPRate (#PCDATA)>
<!ELEMENT MCPErrorStatusCode (#PCDATA)>
<!ELEMENT MCPErrorMessage (#PCDATA)>
```

6.5 MCP Purchase

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

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_purchase>
```

MCP Purchasetransaction object definition

```
<!ELEMENT mcp_purchase (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?, wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional</p> </div>

Variable Name	Type and Limits	Description
		investigation is required

MCP Purchase transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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.
customer ID <cust_id>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits—field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMY date format that is presented on the card.
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring

Variable Name	Type and Limits	Description
		<p>3 – Mail Order / Telephone Order—Instalment</p> <p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
MCP version number <mcp_version>	<p><i>String</i></p> <p>numeric</p> <p>current version is 1.0</p>	Release version number for MCP
cardholder amount <cardholder_amount>	<p><i>String</i></p> <p>12-character numeric</p> <p>smallest discrete unit of foreign currency</p>	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	<p><i>String</i></p> <p>3-character numeric</p>	ISO code representing the foreign currency of the cardholder

MCP Purchase transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor	<i>String</i>	Merchant-defined description sent on

Variable Name	Type and Limits	Description
<dynamic_descriptor>	<p>20-character alphanumeric total of 22 characters including your merchant name and separator</p> <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	<p>a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div> <p>For Pre-Authorization transactions: the value in the dynamic descriptor field will only be carried over to a Pre-Authorization Completion when executing the latter via the Merchant Resource Center; otherwise, the value for dynamic descriptor must be sent again in the Pre-Authorization Completion</p>
wallet indicator	<i>String</i>	
<wallet_indicator>	3-character alphanumeric	<p>Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass.</p> <p>Required for Apple Pay and Google Pay transactions whereby you are using your own API to decrypt the payload</p> <p>Possible values:</p> <p>APP – Apple Pay In-App</p> <p>APW – Apple Pay on the Web</p> <p>GPP – Google Pay™ In-App</p> <p>GPW – Google Pay™ Web</p> <p>VCO – Visa Checkout</p> <p>MMP – Mastercard MasterPass</p>

Variable Name	Type and Limits	Description
<div> NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. </div>		
market indicator <market_indicator>	String 1-character alphabetic	Optional field used by B2B merchants when paying invoices using straight-through processing in order to qualify for lower interchange fees Allowable value is always: J
card match ID <cm_id>	String 50-character alphanumeric	Applies to Offlinx™ only
MCP rate token <mcp_rate_token>	String N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.6 MCP Pre-Authorization

Verifies and locks funds on the customer's credit card. The funds are locked for a specified amount of time based on the card issuer.

To retrieve the funds that have been locked by a Pre-Authorization transaction so that they may be settled in the merchant's account, a Pre-Authorization Completion transaction must be performed. A Pre-Authorization transaction may only be "completed" once.

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_preauth>
```

MCP Pre-Authorization transaction object definition

```
<!ELEMENT mcp_preauth (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?, wallet_indicator?, market_indicator?, cm_id?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

MCP Pre-Authorization transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may

Variable Name	Type and Limits	Description
		<p>have the same order ID.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
customer ID <cust_id>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
credit card number <pan>	<p><i>String</i></p> <p>max 20-character alphanumeric</p>	<p>Credit card number, usually 16 digits—field can be maximum 20 digits in support of future expansion of card number ranges.</p>
expiry date <expdate>	<p><i>String</i></p> <p>4-character alphanumeric</p> <p>YYMM</p>	<p>Expiry date of the credit card, in YYMM format.</p> <div> <p>NOTE: This is the reverse of the MMYM date format that is presented on the card.</p> </div>
electronic commerce indicator <crypt_type>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce</p>

Variable Name	Type and Limits	Description
		<p>indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
MCP version number <mcp_version>	<i>String</i> numeric current version is 1.0	Release version number for MCP
cardholder amount <cardholder_amount>	<i>String</i> 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	<i>String</i> 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Pre-Authorization transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator <div style="border: 1px solid black; padding: 5px; background-color: #e0f7fa;"> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p>

Variable Name	Type and Limits	Description
		<p>NOTE: The 22-character maximum limit must take the "/" into account as one of the characters</p> <p>For Pre-Authorization transactions: the value in the dynamic descriptor field will only be carried over to a Pre-Authorization Completion when executing the latter via the Merchant Resource Center; otherwise, the value for dynamic descriptor must be sent again in the Pre-Authorization Completion</p>
wallet indicator <wallet_indicator>	String 3-character alphanumeric	<p>Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass.</p> <p>Required for Apple Pay and Google Pay transactions whereby you are using your own API to decrypt the payload</p> <p>Possible values:</p> <p>APP – Apple Pay In-App</p> <p>APW – Apple Pay on the Web</p> <p>GPP – Google Pay™ In-App</p> <p>GPW – Google Pay™ Web</p> <p>VCO – Visa Checkout</p> <p>MMP – Mastercard MasterPass</p> <p>NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported.</p>
market indicator <market_indicator>	String 1-character alphabetic	<p>Optional field used by B2B merchants when paying invoices using straight-through processing in order to qualify for lower interchange fees</p> <p>Allowable value is always: J</p>

Variable Name	Type and Limits	Description
card match ID <cm_id>	String 50-character alphanumeric	Applies to Offlinx™ only
MCP rate token <mcp_rate_token>	String N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.7 MCP Pre-Authorization Completion

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

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

<mcp_completion>

MCP Pre-Authorization Completion transaction object definition

```
<!ELEMENT mcp_completion (order_id, txn_number, crypt_type, cust_id, dynamic_descriptor?, ship_indicator?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?) >
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p>

Variable Name	Type and Limits	Description
---------------	-----------------	-------------

Production: <https://www3.moneris.com/mpg/>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

MCP Pre-Authorization Completion transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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>	<i>String</i> 255-character, alphanumeric, hyphens or underscores variable length	Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund) This value is returned in the response of the original transaction Pre-Authorization Completion: references a Pre-Authorization

Variable Name	Type and Limits	Description
Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion		
electronic commerce indicator <crypt_type>	String 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
customer ID <cust_id>	String 50-character alphanumeric	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
	<p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p>	
MCP version number	String	Release version number for MCP

Variable Name	Type and Limits	Description
<mcp_version>	numeric current version is 1.0	
cardholder amount <cardholder_amount>	String 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	String 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Pre-Authorization Completion transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
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

Variable Name	Type and Limits	Description
		<p>issuer as supplemental shipments or completions are pending, submit shipping indicator with a value of P</p> <p>Possible values:</p> <p>P – Partial</p> <p>F – Final</p>
MCP rate token <mcp_rate_token>	String N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.8 MCP Purchase Correction

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

This transaction can be used against a Purchase or Pre-Authorization Completion transaction that occurred same day provided that the batch containing the original transaction remains open.

MCP processing uses the automated closing feature, and Batch Close occurs daily between 10 and 11 pm Eastern Time.

XML transaction object

<mcpurchasecorrection>

MCP Purchase Correction transaction object definition

<!ELEMENT mcp_purchase correction (order_id, txn_number, crypt_type, cust_id)>

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin set-</p>

Variable Name	Type and Limits	Description
		<p>tings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
<p>status check</p> <p><status_check></p>	<p><i>Boolean</i></p> <p>true/false</p>	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

MCP Purchase Correction transaction request fields – Required

Variable Name	Type and Limits	Description
<p>order ID</p> <p><order_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p> <p>a-Z A-Z 0-9 _ - : . @ spaces</p>	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
<p>transaction number</p> <p><txn_number></p>	<p><i>String</i></p> <p>255-character, alphanumeric, hyphens or underscores</p>	<p>Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)</p>

Variable Name	Type and Limits	Description
	variable length	<p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p>
<p>electronic commerce indicator</p> <p><crypt_type></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
<p>customer ID</p> <p><cust_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p>	<p>Merchant-defined field that can be used as an identifier</p>

Variable Name	Type and Limits	Description
	NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Searchable from the Moneris Merchant Resource Center

6.9 MCP Refund

Restores all or part of the funds from a MCP Purchase or MCP Pre-Authorization Completion transaction to the cardholder's card.

Unlike a MCP Purchase Correction, there is a record of both the initial charge and the refund on the cardholder's statement.

For processing refunds on a different card than the one used in the original transaction, the MCP Independent Refund transaction should be used instead.

XML transaction object

<mcp_refund>

MCP Refund transaction object definition

```
<!ELEMENT mcp_refund (order_id, amount, txn_number, crypt_type, cust_id,
dynamic_descriptor?, mcp_version, cardholder_amount, cardholder_currency_code,
mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

MCP Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div> EXAMPLE: 1234567.89 </div>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
transaction number <txn_number>	<i>String</i> 255-character, alphanumeric, hyphens or underscores	Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)

Variable Name	Type and Limits	Description
	variable length	<p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p>
<p>electronic commerce indicator</p> <p><crypt_type></p>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
<p>customer ID</p> <p><cust_id></p>	<p><i>String</i></p> <p>50-character alphanumeric</p>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>

Variable Name	Type and Limits	Description
	NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	
MCP version number <mcp_version>	<i>String</i> numeric current version is 1.0	Release version number for MCP
cardholder amount <cardholder_amount>	<i>String</i> 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	<i>String</i> 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Refund transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
MCP rate token <mcp_rate_token>	<i>String</i> N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction

Variable Name	Type and Limits	Description
---------------	-----------------	-------------

requests in order to redeem that rate

6.10 MCP Independent Refund

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

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

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_ind_refund>
```

MCP Independent Refund transaction object definition

```
<!ELEMENT mcp_ind_refund (order_id, cust_id, pan, expdate, crypt_type, dynamic_descriptor?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required

MCP Independent Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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.
customer ID <cust_id>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date	<i>String</i>	Expiry date of the credit card, in YYMM format.

Variable Name	Type and Limits	Description
<expdate>	4-character alphanumeric YYMM	NOTE: This is the reverse of the MMYM date format that is presented on the card.
MCP version number <mcp_version>	String numeric current version is 1.0	Release version number for MCP
cardholder amount <cardholder_amount>	String 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	String 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Independent Refund transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
MCP rate token <mcp_rate_token>	String N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction

Variable Name	Type and Limits	Description
---------------	-----------------	-------------

requests in order to redeem that rate

6.11 MCP Purchase With Vault

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

The data key may be a temporary one generated used Hosted Tokenization, or may be a permanent one from the Vault.

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_res_purchase_cc>
```

MCP Purchase With Vault transaction object definition

```
<!ELEMENT mcp_res_purchase_cc (data_key, order_id, cust_id, crypt_type,
dynamic_descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>

MCP Purchase With Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>

Variable Name	Type and Limits	Description
	<div>< > \$ % = ? ^ { } [] \</div>	
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <ul style="list-style-type: none"> if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7 if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7 if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7
MCP version number <mcp_version>	<i>String</i> numeric current version is 1.0	Release version number for MCP
cardholder amount <cardholder_amount>	<i>String</i> 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction

Variable Name	Type and Limits	Description
cardholder currency code <cardholder_currency_code>	<i>String</i> 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Purchase With Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. NOTE: This is the reverse of the MMYY date format that is presented on the card.
MCP rate token <mcp_rate_token>	<i>String</i> N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.12 MCP Pre-Authorization With Vault

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

The data key may be a temporary one generated used Hosted Tokenization, or may be a permanent one from the Vault.

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_res_preauth_cc>
```

MCP Pre-Authorization With Vault transaction object definition

```
<!ELEMENT mcp_res_preauth_cc (data_key, order_id, cust_id, crypt_type,
dynamic_descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p>

Variable Name	Type and Limits	Description
		<p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p>

MCP Pre-Authorization With Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	String 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
order ID <order_id>	String 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	<p>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.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
customer ID <cust_id>	String 50-character alphanumeric	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
electronic commerce indicator <crypt_type>	String 1-character alphanumeric	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Instalment

Variable Name	Type and Limits	Description
		<p>4 – Mail Order / Telephone Order—Unknown classification</p> <p>5 – Authenticated e-commerce transaction (3-D Secure)</p> <p>6 – Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
cardholder amount <cardholder_amount>	<p><i>String</i></p> <p>12-character numeric</p> <p>smallest discrete unit of foreign currency</p>	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	<p><i>String</i></p> <p>3-character numeric</p>	ISO code representing the foreign currency of the cardholder

MCP Pre-Authorization With Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	<p><i>String</i></p> <p>20-character alphanumeric</p> <p>total of 22 characters including your merchant name and separator</p>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the</p>

Variable Name	Type and Limits	Description
	NOTE: Some special characters are not allowed: <code>< > \$ % = ? ^ { } [] \</code>	dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYYYMM	Expiry date of the credit card, in YYYYMM format. NOTE: This is the reverse of the MMYM date format that is presented on the card.
MCP rate token <mcp_rate_token>	<i>String</i> N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.13 MCP Independent Refund with Vault

This transaction uses the data key to identify a previously registered credit card profile in Vault. The details saved within the profile are then submitted to perform an Independent Refund transaction.

This transaction request is the multi-currency pricing (MCP) enabled version of the equivalent financial transaction.

XML transaction object

```
<mcp_ind_refund>
```

MCP Independent Refund with Vault transaction object definition

```
<!ELEMENT mcp_res_ind_refund_cc (data_key, order_id, cust_id, crypt_type,
dynamic_descriptor?, mcp_version, cardholder_amount, cardholder_currency_code,
mcp_rate_token?)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	<i>String</i>	Unique identifier provided by Moneris upon merchant account setup

Variable Name	Type and Limits	Description
<store_id>	N/A	
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

MCP Independent Refund with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	String 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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.
customer ID <cust_id>	String 50-character alphanumeric <div>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
electronic commerce indicator <crypt_type>	String 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator , as follows: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values

Variable Name	Type and Limits	Description
		<p>for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
MCP version number <mcp_version>	String numeric current version is 1.0	Release version number for MCP
cardholder amount <cardholder_amount>	String 12-character numeric smallest discrete unit of foreign currency	Amount, in units of foreign currency, the cardholder will be charged on the transaction
cardholder currency code <cardholder_currency_code>	String 3-character numeric	ISO code representing the foreign currency of the cardholder

MCP Independent Refund with Vault transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters including your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div>
MCP rate token <mcp_rate_token>	String N/A	Token representing a temporarily locked-in foreign exchange rate, obtained in the response of the

Variable Name	Type and Limits	Description
		MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

6.14 MCP Get Rate

Performs a foreign currency exchange rate look-up, and secures that exchange rate for use in a subsequent MCP financial transaction.

The exchange rate retrieved by this transaction request is represented in the response as the **RateToken**, and the underlying exchange rate is locked in for a limited time period.

XML transaction object

```
<mcp_get_rate>
```

MCP Get Rate transaction object definition

```
<!ELEMENT mcp_get_rate (mcp_version, rate_txn_type, rate_info)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

Optional connection object field

Variable Name	Type and Limits	Description
status check	Boolean	Checks whether a previously sent

Variable Name	Type and Limits	Description
<status_check>	true/false	<p>transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

MCP Get Rate transaction request fields – Required

Variable Name	Type and Limits	Description
MCP version number <mcp_version>	<i>String</i> numeric current version is 1.0	Release version number for MCP
rate transaction type <rate_txn_type>	<i>String</i> 1-character alphabetic	<p>Value representing the type of subsequent transaction request that the rate token will be used for.</p> <p>Allowable values:</p> <p>P – Purchase</p> <p>R – Refund</p>
MCP Rate Info <rate_info>	<i>Object</i> N/A	Nested object in the MCP Get Rate transaction containing the add cardholder amount and add merchant settlement fields
add cardholder amount	<i>String</i> 12-character numeric, 3-character numeric (smallest discrete unit of foreign currency, currency code)	<p>A string array representing:</p> <ul style="list-style-type: none"> the amount, in units of foreign currency, the cardholder will be charged, and the ISO currency code corresponding to the foreign cur-

Variable Name	Type and Limits	Description
		currency of the cardholder
add merchant settlement amount	<i>String</i> 12-character numeric, 3-character numeric (amount in CAD pennies, currency code)	A string array representing: <ul style="list-style-type: none"> the amount the merchant will receive in the transaction, in Canadian dollars the ISO currency code corresponding to the foreign currency of the cardholder

6.15 MCP Currency Codes

For currency symbols, see <https://justforex.com/education/currencies>

NOTE: This documentation contains links to websites owned and operated by third parties. If you use these links, you leave our website. These links are provided for your information and convenience only and are not an endorsement by Moneris Solutions of the content of such linked websites or third parties. Moneris Solutions has no control over the contents of any linked website and is not responsible for these websites or their content or availability. If you decide to access any third party websites and make use of the information contained on them, you do so entirely at your own risk.

Numeric Currency Code (ISO)	Currency Name/Acronym
008	Albanian Lek (ALL)
012	Algerian Dinar (DZD)
032	Argentine Peso (ARS)
036	Australian Dollar (AUD)
048	Bahraini Dinar (BHD)
050	Bangladeshi Taka (BDT)

Numeric Currency Code (ISO)	Currency Name/Acronym
052	Barbados Dollar (BBD)
060	Bermudian Dollar (BMD)
064	Bhutan Ngultrum (BTN)
068	Bolivia Boliviano (BOB)
084	Belize Dollar (BZD)
090	Solomon Islands Dollar (SBD)
096	Brunei Dollar (BND)
108	Burundi Franc (BIF)
132	Cabo Verde Escudo (CVE)
136	Cayman Islands Dollar (KYD)
144	Sri Lanka Rupee (LKR)
152	Chilean Peso (CLP)
156	Chinese Yuan (CNY)
170	Colombian Peso (COP)
174	Comorian Franc (KMF)
188	Costa Rican Colon (CRC)
191	Croatian Kuna (HRK)
192	Cuban Peso (CUP)
203	Czech Koruna (CZK)
208	Danish Krone (DKK)
214	Dominican Republic Peso
222	Salvadoran Colon (SVC)
242	Fijian Dollar (FJD)

Numeric Currency Code (ISO)	Currency Name/Acronym
262	Djiboutian Franc (DJF)
270	Gambian Dalasi (GMD)
292	Gibraltar Pound (GIP)
320	Guatemala Quetzal (GTQ)
324	Guinean Franc (GNF)
328	Guyanese Dollar (GYD)
332	Haitian Gourde (HTG)
340	Honduran Lempira (HNL)
344	Hong Kong Dollar (HKD)
348	Hungarian Forint (HUF)
352	Iceland Krona (ISK)
356	Indian Rupee (INR)
360	Indonesian Rupiah (IDR)
376	Israeli Shekel (ILS)
388	Jamaican Dollar (JMD)
392	Japanese Yen (JPY)
398	Kazakh Tenge (KZT)
400	Jordanian Dinar (JOD)
404	Kenyan Shilling (KES)
410	South Korean Won (KRW)
414	Kuwaiti Dinar (KWD)
418	Laotian Kip (LAK)
426	Lesotho Loti (LSL)

Numeric Currency Code (ISO)	Currency Name/Acronym
430	Liberian Dollar (LRD)
446	Macanese Pataca (MOP)
454	Malawian Kwacha (MWK)
458	Malaysian Ringgit (MYR)
462	Maldivian Rufiyaa (MVR)
480	Mauritius Rupee (MUR)
484	Mexican Peso (MXN)
498	Moldovan Leu (MDL)
504	Moroccan Dirham (MAD)
512	Omani Rial (OMR)
516	Namibian Dollar (NAD)
524	Nepalese Rupee (NPR)
532	Netherlands Antillean Guilder (ANG)
533	Aruban Guilder (AWG)
548	Vanuatu Vatu (VUV)
554	New Zealand Dollar (NZD)
558	Nicaraguan Cordoba (NIO)
566	Nigerian Naira (NGN)
578	Norwegian Krone (NOK)
586	Pakistan Rupee (PKR)
598	Papua New Guinean Kina (PGK)
600	Paraguayan Guarani (PYG)
604	Peruvian Nuevo Sol (PEN)

Numeric Currency Code (ISO)	Currency Name/Acronym
608	Philippine Peso (PHP)
634	Qatari Rial (QAR)
643	Russian Ruble (RUB)
646	Rwandan Franc (RWF)
654	Saint Helena Pound (SHP)
682	Saudi Riyal (SAR)
690	Seychelles Rupee (SCR)
694	Sierra Leonean Leone (SLL)
702	Singapore Dollar (SGD)
704	Vietnamese Dong (VND)
710	South African Rand (ZAR)
748	Swaziland Lilangeni (SZL)
752	Swedish Krona (SEK)
756	Swiss Franc (CHF)
764	Thai Baht (THB)
780	Trinidad & Tobago Dollar (TTD)
784	UAE Dirham (AED)
788	Tunisian Dinar (TND)
800	Ugandan Shilling (UGX)
807	Macedonian Denar (MKD)
818	Egyptian Pound (EGP)
826	UK Pound Sterling (GBP)
834	Tanzanian Shilling (TZS)

Numeric Currency Code (ISO)	Currency Name/Acronym
840	US Dollar (USD)
858	Uruguayan Peso (UYU)
860	Uzbekistani Sum (UZS)
882	Samoa Tala (WST)
901	New Taiwan Dollar (TWD)
929	Mauritanian Ouguiya (MRU)
933	Belarusian Ruble (BYN)
934	Turkmenistan Manat (TMT)
941	Serbian Dinar (RSD)
943	Mozambique Metical (MZN)
944	Azerbaijani Manat (AZN)
946	Romanian New Leu (RON)
949	New Turkish Lira (TRY)
951	East Caribbean Dollar (XCD)
952	West African CFA Franc BCEAO (XOF)
953	CFP Franc (XPF)
967	Zambian Kwacha (ZMW)
968	Surinamese Dollar (SRD)
969	Malagasy Ariary (MGA)
971	Afghan Afghani (AFN)
972	Tajikistan Somoni (TJS)
973	Angola Kwanza (AOA)
975	Bulgarian Lev (BGN)

Numeric Currency Code (ISO)	Currency Name/Acronym
977	Bosnia and Herzegovina Convertible Mark (BAM)
978	Euro (EUR)
981	Georgian Lari (GEL)
985	Polish New Zloty (PLN)
986	Brazilian Real (BRL)

6.16 MCP Error Codes

Error Code	Description
200	OK (there will be no value returned in the MCP error message)
500	Upstream error
1000	Invalid JSON format
1003	Invalid txnType detected: <invalid txnType> please enter PURCHASE or REFUND
1005	Invalid rateInquiryId-txnType combination.
1007	Warning: at least one of cardHolderCurrency or merchantSettlementCurrency must be non-zero.
1008	Card-holder amount must be non-zero.
1009	Negative amounts detected
1010	Unsupported cardholder currency detected: <unsupported currency>
1015	invalid rateInquiryId
1016	Unsupported merchant id

7 Apple Pay Token Transactions

- 7.1 About Apple Pay Token Transactions
- 7.2 Apple Pay Token Request DTD
- 7.3 Apple Pay Token Purchase
- 7.4 Apple Pay Token Pre-Authorization

7.1 About Apple Pay Token Transactions

ApplePayToken transactions are a list of transactions for merchants who are trying to pass ApplePay PKPayment objects to Moneris for decryption.

This is intended for merchants who are trying to pass the data through their own centralized platform rather than using the mobile device where the InApp transaction takes place. This is an extension of the ApplePay InApp/On the Web transactions. The values used in the transaction can be obtained during the transaction using the PKPayment object returned in Delegate/Events.

Language	Delegate/Event
Objective C	<code>paymentAuthorizationViewController:didAuthorizePayment:handler:</code>
Swift	<code>paymentAuthorizationViewController (_:didAuthorizePayment:handler:)</code>
JavaScript	<code>ApplePaySession.onpaymentauthorized</code>

7.2 Apple Pay Token Request DTD

NOTE: this is an addendum to the Moneris Gateway XML DTD Field definition, the response DTD remains unchanged from other e-commerce transactions.

```
<!-- Main Elements -->
<!ELEMENT request (store_id, api_token, (applepay_token_purchase, applepay_token_preauth))>
<!ELEMENT applepay_token_purchase (order_id, cust_id?, amount, displayName, network, version,
data, signature, header, type, dynamic_descriptor?, token_originator?)>
<!ELEMENT applepay_token_preauth (order_id, cust_id?, amount, displayName, network, version,
data, signature, header, type, dynamic_descriptor?, token_originator?)>
<!ELEMENT header (public_key_hash, ephemeral_public_key, transaction_id)>
<!ELEMENT token_originator (store_id, api_token)>
```

7.3 Apple Pay Token Purchase

This is a purchase transaction used for merchants who want to use the Moneris Unified API to process Apple Pay transactions while also having Moneris handle the decryption.

XML transaction object

```
<applepay_token_purchase>
```

Apple Pay Token Purchase transaction object definition

```
<!ELEMENT applepay_token_purchase (order_id, cust_id?, amount, displayName, network, version, data, signature, header, type, dynamic_descriptor?, token_ originator?)>
```

Apple Pay Token Purchase transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount	String 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
display name <displayName>	String N/A	Field returned by Apple that displays the name of a user's card for ease of recognition
signature <signature>	String N/A	Signature of the payment and header data The signature includes:

Variable Name	Type and Limits	Description
		<ul style="list-style-type: none"> the signing certificate, its intermediate CA certificate, and information about the signing algorithm
data <data>	String N/A	Encrypted payment data, presented as a Base64 Encoded string
version <version>	String N/A	Version information about the payment token Only EC_v1 is supported
header <header>	Object N/A	Additional version-dependent information used to decrypt and verify the payment There are three items in the setter: <ul style="list-style-type: none"> Public Key Hash Ephemeral Public Key Transaction ID

Required Fields for Header Object

Variable Name	Type and Limits	Description
public key hash <public_key_hash>	String N/A	SHA-256 Hash of the X.509 encoded public key bytes of the merchant's certificate
ephemeral public key <ephemeral_public_key>	String N/A	Ephemeral public key bytes

Variable Name	Type and Limits	Description
transaction ID <transaction_id>	String N/A	Transaction identifier, generated on device

Apple Pay Token Purchase transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	String 50-character alphanumeric <div>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
network <network>	String N/A	This field is mandatory for Apple Pay and Google Pay™ INTERAC® e-Commerce transactions whereby the merchant is using their own API to decrypt the payload. Field is case sensitive Possible value: Interac
type <type>	String N/A	This field is mandatory for INTERAC® e-CommerceApple Pay and Google Pay™ transactions whereby the merchant is using their own API to decrypt the payload Field is case sensitive Possible values: 3DSecure = Cryptogram obtained using MerchantCapability3DS EMV = Cryptogram obtained using MerchantCapablitityEMV
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters includ-	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business

Variable Name	Type and Limits	Description
	ing your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div>
token originator <token_originator> <div> NOTE: This object is used for merchants who own multiple merchant accounts and would like to decrypt using the encryption key of a master store </div>	<i>Object</i> N/A	Indicates the master store for the purposes of decryption Applicable for merchants who have multiple merchant accounts but want to decrypt a transaction using the encryption key of a master store

Fields for Optional Token Originator Object

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/

7.4 Apple Pay Token Pre-Authorization

This is a pre-authorization transaction used for merchants who want to use the Moneris Unified API to process Apple Pay transactions while also having Moneris handle the decryption.

XML transaction object

```
<applepay_token_preauth>
```

Apple Pay Token Pre-Authorization transaction object definition

```
<!ELEMENT applepay_token_preauth (order_id, cust_id?, amount, displayName, network, version, data, signature, header, type, dynamic_descriptor?, token_ originator?)>
```

Apple Pay Token Pre-Authorization transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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>	<i>String</i> 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
display name <displayName>	<i>String</i> N/A	Field returned by Apple that displays the name of a user's card for ease of recognition
signature <signature>	<i>String</i> N/A	Signature of the payment and header data The signature includes:

Variable Name	Type and Limits	Description
		<ul style="list-style-type: none"> the signing certificate, its intermediate CA certificate, and information about the signing algorithm
data <data>	String N/A	Encrypted payment data, presented as a Base64 Encoded string
version <version>	String N/A	Version information about the payment token Only EC_v1 is supported
header <header>	Object N/A	Additional version-dependent information used to decrypt and verify the payment There are three items in the setter: <ul style="list-style-type: none"> Public Key Hash Ephemeral Public Key Transaction ID

Required Fields for Header Object

Variable Name	Type and Limits	Description
public key hash <public_key_hash>	String N/A	SHA-256 Hash of the X.509 encoded public key bytes of the merchant's certificate
ephemeral public key <ephemeral_public_key>	String N/A	Ephemeral public key bytes

Variable Name	Type and Limits	Description
transaction ID <transaction_id>	String N/A	Transaction identifier, generated on device

Apple Pay Token Pre-Authorization transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	String 50-character alphanumeric <div>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
network <network>	String N/A	This field is mandatory for Apple Pay and Google Pay™ INTERAC® e-Commerce transactions whereby the merchant is using their own API to decrypt the payload. Field is case sensitive Possible value: Interac
type <type>	String N/A	This field is mandatory for INTERAC® e-CommerceApple Pay and Google Pay™ transactions whereby the merchant is using their own API to decrypt the payload Field is case sensitive Possible values: 3DSecure = Cryptogram obtained using MerchantCapability3DS EMV = Cryptogram obtained using MerchantCapablitityEMV
dynamic descriptor <dynamic_descriptor>	String 20-character alphanumeric total of 22 characters includ-	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business

Variable Name	Type and Limits	Description
	ing your merchant name and separator <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div> <p>For Pre-Authorization transactions: the value in the dynamic descriptor field will only be carried over to a Pre-Authorization Completion when executing the latter via the Merchant Resource Center; otherwise, the value for dynamic descriptor must be sent again in the Pre-Authorization Completion</p>
token originator <token_originator> <div> NOTE: This object is used for merchants who own multiple merchant accounts and would like to decrypt using the encryption key of a master store </div>	<i>Object</i> N/A	Indicates the master store for the purposes of decryption Applicable for merchants who have multiple merchant accounts but want to decrypt a transaction using the encryption key of a master store

Fields for Optional Token Originator Object

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation

Variable Name	Type and Limits	Description
		<p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

8 Recurring Billing

- 8.1 About Recurring Billing
- 8.2 Purchase with Recurring Billing
- 8.3 Recurring Billing Update
- 8.4 Recurring Billing Response Fields and Codes
- 8.5 Credential on File and Recurring Billing

8.1 About Recurring Billing

Recurring Billing allows you to set up payments whereby Moneris automatically processes the transactions and bills customers on your behalf based on the billing cycle information you provide.

Recurring Billing series are created by sending the Recurring Billing object in these transactions:

- Purchase
- Purchase with Vault
- Purchase with 3-D Secure (cavvPurchase)

You can modify a Recurring Billing series after it has been created by sending the Recurring Billing Update administrative transaction.

NOTE: Alternatively, if you prefer to manage recurring series on your own merchant system, you can send the periodic payments as basic Purchase transactions with the e-commerce indicator (`crypt_type`) value = 2 and with the Credential on File info object included.

8.2 Purchase with Recurring Billing

Purchase transaction with the Recurring Billing object included as part of the request.

Recurring Billing allows you to set up payments whereby Moneris automatically processes the transactions and bills customers on your behalf based on the billing cycle information you provide.

Purchase with Recurring Billing transaction object definition

```
<!ELEMENT purchase (order_id, cust_id?, amount, pan, expdate, crypt_type,
dynamic_descriptor?, cust_info?, avs_info?, cvd_info?, recur?, cof_info? )>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	String N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	String N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	Boolean true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Purchase with Recurring Billing transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	String 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may

Variable Name	Type and Limits	Description
		<p>have the same order ID.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
amount <amount>	<p><i>String</i></p> <p>10-character decimal</p> <p>Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point</p> <p>EXAMPLE: 1234567.89</p>	<p>Transaction dollar amount</p> <p>This must contain at least 3 digits, two of which are penny values</p> <p>Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99</p>
credit card number <pan >	<p><i>String</i></p> <p>max 20-character alphanumeric</p>	<p>Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.</p>
expiry date <expdate>	<p><i>String</i></p> <p>4-character alphanumeric</p> <p>YYMM</p>	<p>Expiry date of the credit card, in YYMM format.</p> <p>NOTE: This is the reverse of the MMY date format that is presented on the card.</p>
electronic commerce indicator <crypt_type>	<p><i>String</i></p> <p>1-character alphanumeric</p>	<p>Describes the category of e-commerce transaction being processed. Allowable values are:</p> <ul style="list-style-type: none"> 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 (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure) 7 – SSL-enabled merchant

Variable Name	Type and Limits	Description
		<p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>

Purchase with Recurring Billing transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<p><i>String</i></p> <p>50-character alphanumeric</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p>
dynamic descriptor <dynamic_descriptor>	<p><i>String</i></p> <p>20-character alphanumeric</p> <p>total of 22 characters including your merchant name and separator</p> <div> <p>NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \</p> </div>	<p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> <p>NOTE: The 22-character maximum limit must take the "/" into account as one of the characters</p> </div>

Variable Name	Type and Limits	Description
wallet indicator <wallet_indicator>	<i>String</i> 3-character alphanumeric	<p>Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass.</p> <p>Required for Apple Pay and Google Pay transactions whereby you are using your own API to decrypt the payload</p> <p>Possible values:</p> <p>APP – Apple Pay In-App</p> <p>APW – Apple Pay on the Web</p> <p>GPP – Google Pay™ In-App</p> <p>GPW – Google Pay™ Web</p> <p>VCO – Visa Checkout</p> <p>MMP – Mastercard MasterPass</p> <div> NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. </div>
Customer Information <cust_info>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
<div>For information on request fields for this object, see xrefHere</div>		
AVS Information <avs_info>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
<div>For information on request fields for this object, see xrefHere</div>		
CVD Information <cvd_info>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
<div>For information on request fields for this object, see xrefHere</div>		

Variable Name	Type and Limits	Description
fHere		
Recurring Billing <recur>	<i>Object</i> N/A	Contains fields related to Recurring Billing
For information on request fields for this object, see xrefHere		
Credential on File Information <cof_info>	<i>Object</i> N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
For information on request fields for this object, see xrefHere		

Recurring Billing object request fields

Variable Name	Type and Limits	Description
recur unit <recur_unit>	<i>String</i> day, week, month or eom	Unit to be used as a basis for the interval Works in conjunction with the period variable to define the billing frequency
start now <start_now>	<i>String</i> true/false	Set to true if a charge will be made against the card immediately; otherwise set to false When set to false, use Card Verification prior to sending the Purchase with Recurring Billing and Credential on File objects
NOTE: Amount to be billed immediately can differ from the subsequent recurring amounts		
start date <start_date>	<i>String</i> YYYYMMDD format	Date of the first future recurring billing transaction; this must be a date in the future If an additional charge will be made

Variable Name	Type and Limits	Description
		immediately, the start now variable must be set to true
number of recurs <num_rekurs>	String numeric 1-999	The number of times that the transaction must recur
period <period>	String numeric 1-999	Number of recur unit intervals that must pass between recurring billings
recurring amount <recur_amount>	String 10-character decimal, minimum three digits Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point	Dollar amount of the recurring transaction This amount will be billed on the start date, and then billed repeatedly based on the interval defined by period and recur unit
<div>EXAMPLE: 1234567.89</div>		

8.3 Recurring Billing Update

After you have set up a Recurring Billing transaction series, you can change some of the details of the series as long as it has not yet completed the preset recurring duration (i.e., it hasn't terminated yet).

Before sending a Recurring Billing Update transaction that updates the credit card number, you must send a Card Verification request. This requirement does not apply if you are only updating the schedule or amount.

Things to Consider:

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

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
API token <api_token>	<i>String</i> N/A	<p>Unique alphanumeric string assigned by Moneris upon merchant account activation</p> <p>To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs:</p> <p>Testing: https://esqa.moneris.com/mpg/</p> <p>Production: https://www3.moneris.com/mpg/</p>

Optional connection object field

Variable Name	Type and Limits	Description
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required</p> </div>

Recurring Billing Update transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - . : @ spaces	Merchant-defined transaction identifier that must be unique for every Purchase, Pre-Authorization and Independent Refund transaction. No

Variable Name	Type and Limits	Description
		<p>two transactions of these types may have the same order ID.</p> <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>

Recurring Billing Update transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID <cust_id>	<i>String</i> 50-character alphanumeric <div> NOTE: Some special characters are not allowed: < > \$ % = ? ^ { } [] \ </div>	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
credit card number <pan>	<i>String</i> max 20-character alphanumeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. <div> NOTE: This is the reverse of the MMY Y date format that is presented on the card. </div>
recurring amount <recur_amount>	<i>String</i> 10-character decimal, minimum three digits Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div> EXAMPLE: 1234567.89 </div>	Dollar amount of the recurring transaction This amount will be billed on the start date, and then billed repeatedly based on the interval defined by period and recur unit
add number of recurs	<i>String</i>	Increments a specified number of transactions to the current remain-

Variable Name	Type and Limits	Description
<add_num_rekurs>	3-character numeric 1-999	ing number of recurring transactions Can be used if a customer decides to extend a membership or subscription Must be a positive number, and therefore cannot be used to decrement the number of remaining transactions; for decrementing, use change number of recurs
change number of recurs <total_num_rekurs>	<i>String</i> 3-character numeric 1-999	Replaces the current remaining number of recurring transactions If you only need to increment the number of recurring transactions, use add number of recurs instead
hold recurring billing <hold>	<i>Boolean</i> true/false	Temporarily pauses recurring billing While a transaction is on hold, it is not billed for the recurring amount; however, the number of remaining recurs continues to be decremented during that time
terminate recurring billing <terminate>	<i>Boolean</i> true/false	Terminates recurring billing Once terminated, a recurring billing transaction cannot be reactivated; a new Purchase with Recurring Billing transaction must be submitted
Credential on File Information <cof_info>	<i>Object</i> N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

8.4 Recurring Billing Response Fields and Codes

Table 2 outlines the response fields that are part of recurring billing. Some are available when you set up recurring billing (such as with a Purchase transaction), and some are available when you update an existing transaction with the Recurring Billing transaction.

Receipt object definition

```
Receipt receipt = mpgReq.getReceipt();
```

```
$mpgResponse=$mpgHttpPost->getMpgResponse();
```

```
Receipt receipt = mpgReq.GetReceipt();
```

Table 2: Recurring Billing response fields

Value	Type	Limits	Get method
	Description		
Transaction object with Recurring Billing response fields			
Response code	String	3-character numeric	<code>receipt.getResponseCode () ;</code> <code>\$mpgResponse->getResponseCode ()</code> <code>receipt.GetResponseCode () ;</code>
	See Table 3: for a description of possible response codes.		
Recur success	String	TBD	<code>receipt.getRecurSuccess () ;</code> <code>\$mpgResponse->getRecurSuccess ()</code> <code>receipt.GetRecurSuccess () ;</code>
	Indicates whether the transaction successfully registered		
Recur update object response fields			
Recur update success	String	true/false	<code>receipt.getRecurUpdateSuccess () ;</code> <code>\$mpgResponse->getRecurUpdateSuccess ()</code> <code>receipt.GetRecurUpdateSuccess () ;</code>
	Indicates whether the transaction successfully updated.		
Next recur date	String	yyyy-mm-dd format	<code>receipt.getNextRecurDate () ;</code> <code>\$mpgResponse->getNextRecurDate ()</code> <code>receipt.GetNextRecurDate () ;</code>
	Indicates when the transaction will be billed again.		
Recur end date	String	yyyy-mm-dd format	<code>receipt.getRecurEndDate () ;</code> <code>\$mpgResponse->getRecurEndDate ()</code> <code>receipt.GetRecurEndDate () ;</code>
	Indicates when the Recurring Billing Transaction will end.		

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

Table 3: Recur update response codes

Request Value	Definition
001	Recurring transaction successfully updated (optional: terminated)
983	Cannot find the previous transaction
984	Data error: (optional: field name)
985	Invalid number of recurs
986	Incomplete: timed out
null	Error: Malformed XML

8.5 Credential on File and Recurring Billing

NOTE: The value of the **payment indicator** field must be **R** when sending Recurring Billing transactions.

For Recurring Billing transactions which are set to start **immediately**:

1. Send a Purchase transaction request with both the Recurring Billing and Credential on File info objects (with Recurring Billing object field **start now** = true)

For Recurring Billing transactions which are set to start on a **future** date:

1. Send Card Verification transaction request including the Credential on File info object to get the Issuer ID
2. Send Purchase transaction request with the Recur and Credential on File info objects included

For updating a Recurring Billing series where you are updating the card number (does not apply if you are only modifying the schedule or amount in a recurring series):

1. Send Card Verification request including the Credential on File info object to get the Issuer ID
2. Send a Recurring Billing Update transaction

9 Customer Information

- 9.1 Customer Information Object
- 9.2 Customer Info Object – Billing Information
- 9.3 Customer Info Object – Shipping Information
- 9.4 Customer Information Object – Items

9.1 Customer Information Object

The Customer Information object offers a number of fields to be submitted as part of the financial transaction, and stored by Moneris. These details may be viewed in the future in the Merchant Resource Center.

The following transactions support the Customer Information object :

- Purchase (Basic and Vault)
- Pre-Authorization (Basic and Vault)

XML transaction object

<cust_info>

Customer Information object definition

```
<!-- start Cust Info -->

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



```
<!ELEMENT product_code (#PCDATA)>
<!ELEMENT extended_amount (#PCDATA)>
```

Customer Information object request fields – Required

Variable Name	Type and Limits	Description
billing information <billing>	<i>Object</i> N/A	Sub-object of the customer information object; contains fields related to shipping
shipping information <shipping>	<i>Object</i> N/A	Sub-object of the customer information object; contains fields related to shipping
email <email>	<i>String</i> 60-character alphanumeric	Customer's email address
instructions <instructions>	<i>String</i> 100-character alphanumeric	Instructions or notes

9.2 Customer Info Object – Billing Information

Billing Information and Shipping Information sub-objects contain the same types of request fields, in order to enable different information to be sent for billing and shipping.

Billing Information sub-object request fields

Variable Name	Type and Limits	Description
first name <first_name>	<i>String</i> 30-character alphanumeric	Customer first name
last name <last_name>	<i>String</i> 30-character alphanumeric	Customer last name
company name <company_name>	<i>String</i> 50-character alphanumeric	Customer's company name
address <address>	<i>String</i> 70-character alphanumeric	Customer address
city	<i>String</i>	Customer city

Variable Name	Type and Limits	Description
<city>	30-character alphanumeric	
province/state	<i>String</i>	Customer province or state
<province>	30-character alphanumeric	
postal/ZIP code	<i>String</i>	Customer postal or ZIP code
<postal_code>	30-character alphanumeric	
country	<i>String</i>	Customer's country
<country>	30-character alphanumeric	
phone number	<i>String</i>	Customer's phone number
<phone>	30-character alphanumeric	
fax number	<i>String</i>	Customer fax number
<fax>	30-character alphanumeric	
federal tax	<i>String</i>	Dollar amount of federal tax
<tax1>	10-character alphanumeric	Not used to calculate total amount
provincial/state tax	<i>String</i>	Dollar amount of provincial or state tax
<tax2>	10-character alphanumeric	Not used to calculate total amount
county/local/specialty tax	<i>String</i>	Dollar amount of county, local or specialty tax amount
<tax3>	10-character alphanumeric	Not used to calculate total amount
shipping cost	<i>String</i>	Dollar amount of fees charged for shipping
<shipping_cost>	10-character alphanumeric	Not used to calculate total amount

9.3 Customer Info Object – Shipping Information

Billing Information and Shipping Information sub-objects contain the same types of request fields, in order to enable different information to be sent for billing and shipping.

Shipping Information sub-object request fields

Variable Name	Type and Limits	Description
first name <first_name>	<i>String</i> 30-character alphanumeric	Customer first name
last name <last_name>	<i>String</i> 30-character alphanumeric	Customer last name
company name <company_name>	<i>String</i> 50-character alphanumeric	Sub-object of the customer information object; contains fields related to shipping
address <address>	<i>String</i> 70-character alphanumeric	Customer address
city <city>	<i>String</i> 30-character alphanumeric	Customer city
province/state <province>	<i>String</i> 30-character alphanumeric	Customer province or state
postal/ZIP code <postal_code>	<i>String</i> 30-character alphanumeric	Customer postal or ZIP code
country <country>	<i>String</i> 30-character alphanumeric	Customer's country
phone number <phone>	<i>String</i> 30-character alphanumeric	Customer's phone number
fax number <fax>	<i>String</i> 30-character alphanumeric	Customer fax number
federal tax <tax1>	<i>String</i> 10-character alphanumeric	Dollar amount of federal tax Not used to calculate total amount
province/state <tax2>	<i>String</i> 30-character alphanumeric	Customer province or state

Variable Name	Type and Limits	Description
county/local/specialty tax <tax3>	<i>String</i> 10-character alphanumeric	Dollar amount of county, local or specialty tax amount Not used to calculate total amount
shipping cost <shipping_cost>	<i>String</i> 10-character alphanumeric	Dollar amount of fees charged for shipping Not used to calculate total amount

9.4 Customer Information Object – Items

The Customer Information object can hold information about multiple invoice items, each one represented as their own array object, with the values encapsulated inside an `<item>` tag.

Item array object request fields

Variable Name	Type and Limits	Description
item name <name>	<i>String</i> 45-character alphanumeric	Name of a specific item being purchased
item quantity <quantity>	<i>String</i> 5-character numeric 1-99999	Number of units of a specific item being ordered Must be > 0 or else the item will not be added to the item list
item product code <product_code>	<i>String</i> 20-character alphanumeric	Product code or SKU of an item being purchased
item extended amount <extended_amount>	<i>String</i> 10-character decimal; must contain minimum 3 digits and 2 penny values Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div>	Unit cost, in dollars, multiplied by quantity ordered

10 e-Fraud Tools

- 1 Address Verification Service
- 1 Card Validation Digits (CVD)

10.1 Address Verification Service (AVS)

- 10.1.1 About Address Verification Service (AVS)
- 10.1.2 AVS Information Object
- 10.1.3 AVS Response Codes

10.1.1 About Address Verification Service (AVS)

Address Verification Service (AVS) is an optional fraud-prevention tool offered by issuing banks whereby a cardholder's address is submitted as part of the transaction authorization. The AVS address is then compared to the address kept on file at the issuing bank. AVS checks whether the street number, street name and zip/postal code match. The issuing bank returns an AVS result code indicating whether the data was matched successfully. Regardless of the AVS result code returned, the credit card is authorized by the issuing bank.

The response that is received from AVS verification is intended to provide added security and fraud prevention, but the response itself does not affect the completion of a transaction. Upon receiving a response, the choice to proceed with a transaction is left entirely to the merchant. The responses is **not** a strict guideline of whether a transaction will be approved or declined.

The following transactions support AVS:

- Purchase (Basic and Mag Swipe)
- Pre-Authorization (Basic)
- Re-Authorization (Basic)
- ResAddCC (Vault)
- ResUpdateCC (Vault)

Things to Consider:

- AVS is supported by Visa, MasterCard, American Express, Discover and JCB.
- When testing AVS, you must **only** use the Visa test card numbers 4242424242424242 or 4005554444444403, and the amounts described in the Simulator eFraud Response Codes document available at the Moneris developer portal (<https://developer-moneris.com>).
- Store ID “store5” is set up to support AVS testing.

10.1.2 AVS Information Object

Contains fields applying to the Address Verification Service (AVS) e-fraud tool

XML transaction object

```
<avs_info>
```

AVS Information Object object definition

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

```
<!ELEMENT avs_street_number (#PCDATA)>
```

```
<!ELEMENT avs_street_name (#PCDATA)>
```

```
<!ELEMENT avs_zipcode (#PCDATA)>
```

AVS Information Object request fields – Required

Variable Name	Type and Limits	Description
AVS street number <avs_street_name>	<i>String</i> 19-character alphanumeric	Cardholder's address street number
AVS street name <avs_street_name>	<i>String</i> 19-character alphanumeric	Cardholder's address street name
AVS postal/ZIP code <xmlvariableHere>	<i>String</i> 9-character alphanumeric	Cardholder's address postal or ZIP code

10.1.3 AVS Response Codes

Below is a full list of possible AVS response codes.

Code	Visa	Mastercard/Discover	American Express/ JCB
A	Street address matches, zip/postal code does not; acquirer rights not implied	Address matches, zip/postal code does not	Billing address matches, zip/postal code does not
B	Street address matches; zip/postal code not verified due to incompatible formats (acquirer sent both street address and zip/postal code)	N/A	N/A
C	Street address not verified due to incompatible formats (acquirer sent both street address and zip/postal code)	N/A	N/A
D	Street address and zip/-postal code match	N/A	Customer name incorrect; zip/postal code matches
E	N/A	N/A	Customer name incorrect, billing address and zip/-postal code match
F	<i>Applies to UK only:</i> Street address and zip/postal code match	N/A	Customer name incorrect; billing address matches
G	Address information not verified for international transaction Any of following may be true: <ul style="list-style-type: none"> • Issuer is not an AVS participant, or • AVS data was present in the request but issuer 	N/A	N/A

Code	Visa	Mastercard/Discover	American Express/ JCB
	<p>did not return an AVS result, or</p> <ul style="list-style-type: none"> • Visa performs AVS on behalf of the issuer and there was no address record on file for this account 		
I	Address information not verified	N/A	N/A
K	N/A	N/A	Customer name matches
L	N/A	N/A	Customer name and zip/-postal code match
M	Street address and zip/-postal code match	N/A	Customer name, billing address, and zip/postal code match
N	<p>No match; acquirer sent:</p> <ul style="list-style-type: none"> • postal/ZIP code only, or • street address only, or • both postal code and street address <p>Also used when acquirer requests AVS but sends no AVS data</p>	Neither address nor zip/-postal code matches	Billing address and zip/-postal code do not match
O	N/A	N/A	Customer name and billing address match
P	Zip/postal code match;	N/A	N/A

Code	Visa	Mastercard/Discover	American Express/ JCB
	acquirer sent both zip/-postal code and street address, but street address not verified due to incompatible formats		
R	Retry; system unavailable or timed out Issuer ordinarily performs AVS, but was unavailable <div> NOTE: Code R is used by Visa when issuers are unavailable; issuers should refrain from using this code. </div>	Retry; system unable to process	System unavailable; retry
S	N/A	AVS currently not supported	AVS currently not supported
T	N/A	Nine-digit zip code matches; address does not match	N/A
U	Address not verified for domestic transaction, for any of the following reasons: <ul style="list-style-type: none"> • Issuer is not an AVS participant, or • AVS data was present in the request but issuer did not return an AVS result, or • Visa performs AVS on behalf of the issuer and there was no address record on file for this 	No data from issuer-authorization system	Information is unavailable

Code	Visa	Mastercard/Discover	American Express/ JCB
account			
W	Not applicable; if present, replaced with Z by Visa <i>Available for U.S. issuers only</i>	For U.S. addresses, nine-digit postal code matches, address does not For addresses outside the U.S., postal code matches, address does not	Customer name, billing address, and zip/postal code are all correct matches
X	N/A	For U.S. addresses, nine-digit postal code and address match For addresses outside the U.S., postal code and address match	N/A
Y	Street address and zip/-postal code match	Billing address and zip/-postal code both match	Billing address and zip/-postal code both match
Z	Zip/postal code matches; street address does not match, or street address not included in request	For U.S. addresses, five-digit zip code matches, address does not match	Zip/postal code matches, billing address does not

10.2 Card Validation Digits (CVD)

- 10.2.1 About Card Validation Digits (CVD)
- 10.2.2 Transactions Where CVD Is Required
- 10.2.3 CVD Information Object
- 10.2.4 CVD Result Codes

10.2.1 About Card Validation Digits (CVD)

The Card Validation Digits (CVD) value is an additional number printed on credit cards that is used as an additional check when verifying cardholder credentials during a transaction.

The response that is received from CVD verification is intended to provide added security and fraud prevention, but the response itself does not affect the completion of a transaction. Upon receiving a

response, the choice whether to proceed with a transaction is left entirely to the merchant. The responses is **not** a strict guideline of which transaction will approve or decline.

The following transactions support CVD:

- Purchase (Basic, Vault and Mag Swipe)
- Pre-Authorization (Basic and Vault)
- Re-Authorization

Things to Consider:

- CVD is only supported by Visa, MasterCard, American Express, Discover, JCB and UnionPay.
- For UnionPay cards, the CVD response will not be returned; the issuer will approve or decline based on the CVD result.
- When testing CVD, you must **only** use the Visa test card numbers 4242424242424242 or 4005554444444403, and the amounts described in the Simulator eFraud Response Codes document available at the Moneris developer portal (<https://developer-moneris.com>).
- Test store_id “store5” is set up to support CVD testing.

10.2.2 Transactions Where CVD Is Required

The Card Validation Digits (CVD) object is required in transaction requests in the following scenarios:

- Initial transactions when storing cardholder credentials in Credential on File scenarios; subsequent follow-on transactions do not use CVD
- Any Purchase, Pre-Authorization or Card Verification where you are not storing cardholder credentials

10.2.3 CVD Information Object

CVD Information object definition

```
<!-- start CVD -->

<!ELEMENT cvd_info (cvd_indicator, cvd_value)>

<!ELEMENT cvd_indicator (#PCDATA)>

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

CVD Info object request fields – Required

Variable Name	Type and Limits	Description
CVD indicator <cvd_indicator>	String 1-character numeric	Indicates presence of CVD Possible values: 0 – CVD value is deliberately bypassed or is not provided by the merchant 1 – CVD value is present 2 – CVD value is printed on the card, but is illegible 9 – Cardholder states that the card has no CVD
CVD value <cvd_value>	String 4-character numeric	CVD value printed on card NOTE: The CVD value must only be passed to the Moneris Gateway. Under no circumstances may it be stored for subsequent uses or displayed as part of the receipt information.

10.2.4 CVD Result Codes

CVD verification is available for Visa, Mastercard, Discover, American Express, JCB and UnionPay transactions.

Code	Description
M	Match
N	No match
P	Not processed
S	CVD should be on the card, but Merchant has indicated that CVD is not present
U	Issuer is not a CVD participant
Y	Match for American Express/JCB only
D	Invalid security code for American Express or JCB only

Code	Description
Other	Invalid response code

11 Definition of Request Fields

Core Request Fields

Variable Name	Type and Limits	Description
amount <amount>	String 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point EXAMPLE: 1234567.89	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = \$0.01, maximum allowable value = \$9999999.99
API token <api_token>	String N/A	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin settings in the Merchant Resource Center, at the following URLs: Testing: https://esqa.moneris.com/mpg/ Production: https://www3.moneris.com/mpg/
authorization code <auth_code>	String 8-character alphanumeric	An authorization code required to carry out a Force Post; provided in the transaction response from the issuing bank
completion amount <comp_amount>	String 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point EXAMPLE: 1234567.89	Dollar amount of a Pre-Authorization Completion transaction, which may differ from the original amount authorized in the Pre-Authorization

Variable Name	Type and Limits	Description
credit card number <pan>	<i>String</i> max 20-character alpha-numeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
customer ID <cust_id>	<i>String</i> 50-character alphanumeric NOTE: Some special characters are not allowed: <>\$%=?^{}[]\	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
dynamic descriptor <dynamic_descriptor>	<i>String</i> 20-character alphanumeric total of 22 characters including your merchant name and separator NOTE: Some special characters are not allowed: <>\$%=?^{}[]\	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters
electronic commerce indicator <crypt_type>	<i>String</i> 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Installment 4 – Mail Order / Telephone Order—Unknown classification 5 – Authenticated e-commerce transaction (3-D Secure) 6 – Non-authenticated e-commerce transaction (3-D Secure)

Variable Name	Type and Limits	Description
		<p>7 – SSL-enabled merchant</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent: the allowable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
expiry date <expdate>	<i>String</i> 4-character alphanumeric YYYY	Expiry date of the credit card, in YYMM format. <div> NOTE: This is the reverse of the MMY Y date format that is presented on the card. </div>
order ID <order_id>	<i>String</i> 50-character alphanumeric-Z A-Z 0-9 _ - : . @ spaces	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. <p>For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.</p>
original order ID <orig_order_id>	<i>String</i> 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	Order ID from the original Pre-Authorization transaction, used as a reference to retrieve the original payment details
shipping indicator <ship_indicator>	<i>String</i> 1-character alphanumeric	Used to identify completion transactions that require multiple shipments, also referred to as multiple completions

Variable Name	Type and Limits	Description
		<p>By default, if shipping indicator is not sent, the Pre-Authorization Completion is listed as final</p> <p>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</p> <p>Possible values:</p> <p>P – Partial</p> <p>F – Final</p>
status check <status_check>	<i>Boolean</i> true/false	<p>Checks whether a previously sent transaction was processed successfully</p> <p>To send a status check request, resend the original transaction with all the same request parameter values, except with status check = true</p> <div> NOTE: Only use once per transaction and within two minutes of the original transaction request; if the status check request times out, do not send again, as additional investigation is required </div>
store ID <store_id>	<i>String</i> N/A	Unique identifier provided by Moneris upon merchant account setup
transaction number <txn_number>	<i>String</i> 255-character, alpha-numeric, hyphens or underscores variable length	<p>Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)</p> <p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Author-</p>

Variable Name	Type and Limits	Description
		ization Completion
wallet indicator <wallet_indicator>	<i>String</i> 3-character alphanumeric	<p>Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay, Visa Checkout and Mastercard MasterPass.</p> <p>Required for Apple Pay and Google Pay transactions whereby you are using your own API to decrypt the payload</p> <p>Possible values:</p> <p>APP – Apple Pay In-App</p> <p>APW – Apple Pay on the Web</p> <p>GPP – Google Pay™ In-App</p> <p>GPW – Google Pay™ Web</p> <p>VCO – Visa Checkout</p> <p>MMP – Mastercard MasterPass</p> <p>NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported.</p>

11.1 Definition of Request Fields – Admin Transactions

Variable Name	Type and Limits	Description
electronic cash register (ECR) number <xmlvariableHere>	<i>String</i> N/A	Identification number assigned to a particular electronic cash register; provided by Moneris

11.2 Definition of Request Fields – Vault

Request fields applicable to Vault transactions

Variable Name	Type and Limits	Description
data key <data_key>	<i>String</i> 25-character alphanumeric	Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered
data key format <data_key_format>	<i>String</i> 2-character alphanumeric	Specifies the data key format being returned If left blank, data key format will default to 25-character alphanumeric Possible values: 0 – 25 character alphanumeric data key 0U – unique 25-character alphanumeric data key
duration <duration>	<i>String</i> 3-character numeric maximum 900 seconds	Amount of time the temporary token should be available
email address <email>	<i>String</i> 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	<i>String</i> 30-character alphanumeric	Used for any supplementary information related to the customer Can be sent in when creating or updating a Vault profile
phone number <phone>	<i>String</i> 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile

11.3 Definition of Request Fields – 3-D Secure 2.0

Variable Name	Type and Limits	Description
billing address	<i>String</i>	Cardholder billing address

Variable Name	Type and Limits	Description
billing city	50-character alphanumeric	Cardholder billing city
	<i>String</i> 50-character alphanumeric	
billing country	3-character alphanumeric	Defined as 3 digit country code ISO 3166-1
	<i>String</i> 3-character alphanumeric	
billing postal code	16-character alphanumeric	Cardholder billing postal code
	<i>String</i> 16-character alphanumeric	
billing province	3-character alphanumeric	Defined in country subdivision ISO 3166-2
	<i>String</i> 3-character alphanumeric	
browser java enabled	1-character alphabetic	Indicates whether Java is enabled in the browser Allowable values: T = True F = False
	<i>String</i> 1-character alphabetic	
browser language	8-character alphanumeric	As defined in IETF BCP47
	<i>String</i> 8-character alphanumeric	
browser screen height	6-character numeric	Pixel height of cardholder screen
	<i>String</i> 6-character numeric	
browser screen width	6-character numeric	Pixel width of cardholder screen
	<i>String</i> 6-character numeric	
browser user agent	2048-character alpha-numeric	Browser User Agent
	<i>String</i> 2048-character alpha-numeric	
cardholder name	45-character alphanumeric	Name of the cardholder
	<i>String</i> 45-character alphanumeric	
challenge window size	2-character alphanumeric	Relates to the rendering of the ACS challenge within the browser.
	<i>String</i> 2-character alphanumeric	

Variable Name	Type and Limits	Description
		<p>Allowable values:</p> <p>01 = 250 x 400</p> <p>02 = 390 x 400</p> <p>03 = 500 x 600</p> <p>04 = 600 x 400</p> <p>05 = Full screen</p>
credit card number	<i>String</i> max 20-character alpha-numeric	Credit card number, usually 16 digits —field can be maximum 20 digits in support of future expansion of card number ranges.
cres	<i>String</i> 200-character alpha-numeric	Response data from the challenge
currency	<i>String</i> 3-character numeric	ISO 4217 3 digit currency code (CAD = 124, USD = 840)
NOTE: This field should not be sent unless Multi Currency Pricing is enabled on your merchant account		
data key	<i>String</i> 25-character alphanumeric	<p>Unique identifier for a Vault profile, and used in future Vault financial transactions to associate a transaction with that profile</p> <p>Data key is generated by Moneris and returned to you in the Receipt object when the profile is first registered</p>
email	<i>String</i> 254-character alpha-numeric	Cardholder email address
expiry date	<i>String</i> 4-character alphanumeric YYMM	<p>Expiry date of the credit card, in YYMM format.</p> <p>NOTE: This is the reverse of the MMY date format that is presented on the card.</p>

Variable Name	Type and Limits	Description
notification URL	<i>String</i> 256-character alpha-numeric	Website URL that will receive 3DS Method Completion response from ACS
request challenge	<i>String</i> 1-character alphabetic	Indicates whether a challenge is requested for this transaction Allowable values: Y = Yes N = No
request type	<i>String</i> 2-character alphanumeric	Allowable values: 01 = cardholder initiated payment 02 = recur
shipping address	<i>String</i> 50-character alphanumeric	Shipping destination address
shipping city	<i>String</i> 50-character alphanumeric	Shipping destination city
ship country	<i>String</i>	Shipping destination country Defined as 3-digit country code in ISO 3166-1
shipping postal code	<i>String</i> 16-character alphanumeric	Shipping destination postal code
shipping province	<i>String</i> 3-character alphanumeric	Shipping destination province Defined in country subdivision ISO 3166-2
3DS completion indicator	<i>String</i> 1-character alphanumeric	indicates whether 3ds method MpiCardLookup was successfully completed Allowable values: Y = Successfully completed N = Did not successfully complete U = Unavailable

11.4 Definition of Request Fields – Information Objects

Information objects are nested objects within transactions that provide additional features, in transactions where they are applicable.

For each object, there are additional request fields contained within the object.

Information object request fields – top level

Variable Name	Type and Limits	Description
AVS Information <avs_info>	<i>Object</i> N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
Credential on File Information <cof_info>	<i>Object</i> N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Convenience Fee Information <convfee_info>	<i>Object</i> N/A	Contains fields related to the Convenience Fee feature
Customer Information <cust_info>	<i>Object</i> N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
CVD Information <cvd_info>	<i>Object</i> N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur>	<i>Object</i> N/A	Contains fields related to Recurring Billing

11.5 Definition of Request Fields – Credential on File

Variable Name	Type and Limits	Description
issuer ID <issuer_id>	<i>String</i> 15-character alphanumeric variable length	Unique identifier for the cardholder's stored credentials Sent back in the response from the

Variable Name	Type and Limits	Description
		<p>card brand when processing a Credential on File transaction</p> <p>If the cardholder's credentials are being stored for the first time, and the issuer ID was returned in the response, you must save the issuer ID on your system to use in subsequent Credential on File transactions (applies to merchant-initiated transactions only)</p> <p>The issuer ID must be saved to your systems when returned from Moneris Gateway in the response data, regardless if the value was received or not</p> <p>As a best practice, if the issuer ID is not returned and you received a value of NULL instead, store that value and send it in the subsequent transaction</p>
payment indicator	<i>String</i>	Indicates the current or intended use of the credentials
<payment_indicator>	1-character alphabetic	<p>Possible values for first transactions:</p> <p>C - unscheduled Credential on File (first transactions only)</p> <p>R - recurring</p> <p>Possible values for subsequent transactions:</p> <p>R - recurring</p> <p>U - unscheduled merchant-initiated transaction</p> <p>Z - unscheduled customer-initiated transaction</p> <p>In Credential on File transactions where the request field e-commerce indicator is also being sent, the acceptable values for e-commerce indicator are dependent on the value sent for payment indicator, as follows:</p> <p>if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6</p> <p>if payment indicator = C, then allowable values</p>

Variable Name	Type and Limits	Description
		<p>for e-commerce indicator: 1, 5, 6 or 7</p> <p>if payment indicator = U, then allowable values for e-commerce indicator: 1 or 7</p> <p>if payment indicator = Z, then allowable values for e-commerce indicator: 1, 5, 6 or 7</p>
payment information <payment_information>	<i>String</i> 1-character numeric	<p>Describes whether the transaction is the first or subsequent in the series</p> <p>Possible values:</p> <p>0 - first transaction in a series (storing payment details provided by the cardholder)</p> <p>2 - subsequent transactions (using previously stored payment details)</p>

11.6 Definition of Request Fields – Apple Pay Token

Apple Pay Token transaction request fields – Required

Variable Name	Type and Limits	Description
display name <displayName>	<i>String</i> N/A	Field returned by Apple that displays the name of a user's card for ease of recognition
signature <signature>	<i>String</i> N/A	<p>Signature of the payment and header data</p> <p>The signature includes:</p> <ul style="list-style-type: none"> the signing certificate, its intermediate CA certificate, and information about the signing algorithm
data <data>	<i>String</i> N/A	Encrypted payment data, presented as a Base64 Encoded string

Variable Name	Type and Limits	Description
version	<i>String</i>	Version information about the payment token
<version>	N/A	Only EC_v1 is supported
header	<i>Object</i>	Additional version-dependent information used to decrypt and verify the payment.
<header>	N/A	There are three items in the setter: Public Key Hash, Ephemeral Public Key, Transaction ID
public key hash	<i>String</i>	SHA-256 Hash of the X.509 encoded public key bytes of the merchant's certificate
<public_key_hash>	N/A	
ephemeral public key	<i>String</i>	Ephemeral public key bytes
<ephemeral_public_key>	N/A	
transaction ID	<i>String</i>	Transaction identifier, generated on device
<transaction_id>	N/A	

Apple Pay Token transaction request fields – Optional

Variable Name	Type and Limits	Description
network	<i>String</i>	Description of the payment network to be used, contains the string representation of the PKPayment.paymentMethod.network. This field is required for ApplePay INTERAC transactions
<network>	N/A	This field is mandatory for Apple Pay and Google Pay™ INTERAC® e-Commerce transactions whereby the merchant is using their own API to decrypt the payload. Field is case sensitive Possible value: Interac

Variable Name	Type and Limits	Description
type	<i>String</i>	Description of the payment method type, contains the string representation of the PKPayment.paymentMethod.type. This field is required for ApplePay INTERAC transactions
<type>	N/A	<p>This field is mandatory for INTERAC® e-CommerceApple Pay and Google Pay™ transactions whereby the merchant is using their own API to decrypt the payload</p> <p>Field is case sensitive</p> <p>Possible values:</p> <p>3DSecure = Cryptogram obtained using MerchantCapability3DS</p> <p>EMV = Cryptogram obtained using MerchantCapablitivyEMV</p>
token originator	<i>Object</i>	This is used for merchants who owns multiple merchant accounts and would like to decrypt using the encryption key of a master store
<token_originator>	N/A	

11.7 Definition of Request Fields – Recurring Billing

Recurring Billing Info Object Request Fields

Variable Name	Type and Limits	Description
number of recurs	<i>String</i>	The number of times that the transaction must recur
<num_recur>	numeric 1-999	
period	<i>String</i>	Number of recur units that must pass between recurring billings
<period>	numeric 1-999	
recurring amount	<i>String</i>	Dollar amount of the recurring transaction
<recur_amount>	10-character decimal, minimum three digits	

Variable Name	Type and Limits	Description
	Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point EXAMPLE: 1234567.89	This amount will be billed on the start date, and then billed repeatedly based on the interval defined by period and recur unit
recur unit <recur_unit>	String day, week, month or eom	Unit to be used as a basis for the interval Works in conjunction with the period variable to define the billing frequency
start date <start_date>	String YYYYMMDD format	Date of the first future recurring billing transaction; this must be a date in the future If an additional charge will be made immediately, the start now variable must be set to true
start now <start_now>	String true/false	Set to true if a charge will be made against the card immediately; otherwise set to false When set to false, use Card Verification prior to sending the Purchase with Recurring Billing and Credential on File objects NOTE: Amount to be billed immediately can differ from the subsequent recurring amounts

11.8 Definition of Request Fields – AVS Info Object

Request fields within the Address Verification Service (AVS) Information object

Variable Name	Type and Limits	Description
AVS postal/ZIP code <avs_zipcode>	String 9-character alphanumeric	Cardholder's address postal or ZIP code

Variable Name	Type and Limits	Description
AVS street name <avs_street_name>	<i>String</i> 19-character alphanumeric	Cardholder's address street name
AVS street number <avs_street_number>	<i>String</i> 19-character alphanumeric	Cardholder's address street number

11.9 Definition of Request Fields – CVD Info Object

Request fields within the Card Validation Digits (CVD) Information object

CVD Info object request fields – Required

Variable Name	Type and Limits	Description
CVD indicator <cvd_indicator>	<i>String</i> 1-character numeric	Indicates presence of CVD Possible values: 0 – CVD value is deliberately bypassed or is not provided by the merchant 1 – CVD value is present 2 – CVD value is printed on the card, but is illegible 9 – Cardholder states that the card has no CVD
CVD value <cvd_value>	<i>String</i> 4-character numeric	CVD value printed on card NOTE: The CVD value must only be passed to the Moneris Gateway. Under no circumstances may it be stored for subsequent uses or displayed as part of the receipt information.