Transaction DTD Merchant Integration Guide

Version: 1.6.9

Copyright © Moneris Solutions, 2025

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 **6**

Changes in v1.6.9 ..... **7**

1 About the Moneris Gateway Transaction DTD **12**
1.1 DTD and XML Syntax 12
1.2 Server to Server Endpoints 12

2 Basic Transaction Set **13**
2.1 Process Flow for Basic Transactions 14
2.2 Basic Request DTD 14
2.3 Basic Response DTD 20
2.4 Purchase 22
2.5 Pre-Authorization 28
2.6 Incremental Pre-Authorization 34
2.7 Re-Authorization 36
2.8 Pre-Authorization Completion 40
2.9 Force Post 43
2.10 Refund 47
2.11 Independent Refund 51
2.12 Purchase Correction 55
2.13 Card Verification 58
2.14 Batch Close 62
2.15 Open Totals 64

3 Credential on File **66**
3.1 About Credential on File 66
3.2 Credential on File Info Object and Variables 66
3.3 Credential on File Transaction Types 67
3.4 Initial Transactions in Credential on File 67
3.5 Credential on File and Converting Temporary Tokens 68
3.6 Card Verification and Credential on File Transactions 68
3.6.1 When to Use Card Verification With COF 69
3.6.2 Credential on File and Vault Add Token 69
3.6.3 Credential on File and Vault Update Credit Card 69
3.6.4 Credential on File and Vault Add Credit Card 69
3.6.5 Credential on File and Recurring Billing 70
0.1 Definition of Request Fields – Credential on File 70

4 Vault Transaction Set **73**
4.1 About the Vault Transaction Set 73
4.2 Vault Request DTD 73
4.3 Vault Response DTD 75
4.4 Vault and Installments 76
4.5 Vault Administrative Transactions 76
4.5.1 Vault Add Credit Card – res_add_cc 76
4.5.2 Vault Update Credit Card – res_update_cc 80
4.5.3 Vault Delete – res_delete 84
4.5.4 Vault Tokenize Credit Card - res_tokenize_cc 85
4.5.5 Vault Lookup Full – res_lookup_full 89
4.5.6 Vault Lookup Masked – res_lookup_masked 91

4.5.7 Vault Is Corporate Card – res_iscorporatecard ... 92
4.5.8 Vault Get Expiring – res_get_expiring ... 94
4.5.9 Vault Temporary Token Add – res_temp_add ... 95
4.5.10 Vault Add Token – res_add_token ... 98
4.6 Vault Financial Transactions ... 102
4.6.1 Customer ID Changes ... 102
4.6.2 Purchase with Vault – res_purchase_cc ... 103
4.6.3 Pre-Authorization with Vault – res_preauth_cc ... 107
4.6.4 Independent Refund with Vault – res_ind_refund_cc ... 111
4.6.5 Force Post with Vault – res_forcepost_cc ... 114
4.6.6 Card Verification with Vault – res_card_verification_cc ... 118

5 3-D Secure 2.2 Transactions ... 122
5.1 About 3-D Secure 2.2 ... 122
5.1.1 3-D Secure Implementations ... 123
5.1.2 Out of Scope/Not Supported Check ... 124
5.1.3 Version Compatibility ... 124
5.1.4 Upgrading from 3-D Secure 2.0 to 3-D Secure 2.2 Check ... 124
5.2 Building Your 3-D Secure 2.2 Integration ... 125
5.2.1 Activating 3-D Secure Functionality ... 125
5.2.2 Transaction Flow for 3-D Secure - Browser channel ... 125
5.2.3 Transaction Flow for 3-D Secure - 3RI channel ... 127
5.2.3.1 Decoupled Authentication ... 128
5.2.4 Mpi2Request Object and 3-D Secure Authentication ... 128
5.2.5 3-D Secure 2.2 Request DTD ... 128
5.2.6 3-D Secure 2.2 Response DTD ... 129
5.3 Implementing Card Lookup Request ... 130
5.3.1 Card Lookup Request ... 130
5.4 Handling the 3DS Method for Device Fingerprinting ... 133
5.5 Implementing MPI 3DS Authentication Request ... 133
5.5.1 MPI 3DS Authentication Request - Browser Channel ... 134
5.5.2 MPI 3DS Authentication Request - 3RI with recurring ... 145
5.5.3 MPI 3DS Authentication Request - 3RI, non-recurring ... 156
5.6 Handling the Challenge Flow ... 164
5.6.1 Cavv Lookup Request ... 165
5.7 Handling the Decoupled Authentication Flow ... 166
5.8 Performing the Authorization ... 167
5.8.1 Purchase with 3-D Secure – cavv_purchase ... 167
5.8.2 Pre-Authorization with 3-D Secure – cavv_preauth ... 173
5.9 Testing Your 3-D Secure 2.2 Integration ... 179
5.10 Moving to Production With 3-D Secure 2.2 ... 179
5.11 3-D Secure 2.2 TransStatus Codes ... 180
5.12 3-D Secure 2.2 Commons TransStatusReason Decline Codes ... 180
5.13 CAVV Result Codes ... 182
5.13.1 Visa CAVV Result Codes ... 182
5.13.2 Mastercard CAVV Result Codes ... 183
5.13.3 American Express CAVV Result Codes ... 184

6 Installments by Visa ... 186
6.1 About Installments by Visa ... 186
6.2 Installments by Visa Transaction Types ... 186
6.3 Sending Transactions with Installments by Visa ... 187
6.4 Installment Plan Lookup ... 187

6.5 Vault Installment Plan Lookup 189
6.6 Installment Info Object 190

7 Multi-Currency Pricing (MCP) 191
7.1 About Multi-Currency Pricing (MCP) 191
7.2 Methods of Processing MCP Transactions 192
7.3 Multi-Currency Pricing (MCP) Request DTD 192
7.4 Multi-Currency Pricing (MCP) Response DTD 193
7.5 MCP Purchase 193
7.6 MCP Purchase with 3-D Secure 198
7.7 MCP Purchase with 3-D Secure and Vault 205
7.8 MCP Pre-Authorization 210
7.9 MCP Pre-Authorization with 3-D Secure 215
7.10 MCP Pre-Authorization with 3-D Secure and Vault 221
7.11 MCP Pre-Authorization Completion 227
7.12 MCP Purchase Correction 231
7.13 MCP Refund 234
7.14 MCP Independent Refund 238
7.15 MCP Purchase With Vault 242
7.16 MCP Pre-Authorization With Vault 246
7.17 MCP Independent Refund with Vault 250
7.18 MCP Get Rate 254
7.19 MCP Currency Codes 256
7.20 MCP Error Codes 262

8 Apple Pay Token Transactions 264
8.1 About Apple Pay Token Transactions 264
8.2 Apple Pay Token Request DTD 264
8.3 Apple Pay Token Purchase 265
8.4 Apple Pay Token Pre-Authorization 269

9 Google Pay Transactions 274
9.1 About Google Pay Transactions 274
9.2 Google Pay™ Transaction Process Overview 274
9.3 Google Pay Purchase 280
9.4 Google Pay Pre-Authorization 283
9.5 Google Pay Temporary Token Add 286
9.6 Google Pay Token Purchase 291
9.7 Google Pay Token Preauth 296

10 Recurring Billing 302
10.1 About Recurring Billing 302
10.2 Purchase with Recurring Billing 302
10.3 Recurring Billing Update 309
10.4 Recurring Billing Response Fields and Codes 312
10.5 Credential on File and Recurring Billing 314

11 Customer Information 315
11.1 Customer Information Object 315
11.2 Customer Info Object – Billing Information 316
11.3 Customer Info Object – Shipping Information 318
11.4 Customer Information Object – Items 319

12 e-Fraud Tools 321
12.1 Address Verification Service (AVS) 321

12.1.1 About Address Verification Service (AVS) 321
12.1.2 AVS Information Object 322
12.1.3 AVS Response Codes 323
12.2 Card Validation Digits (CVD) 324
12.2.1 About Card Validation Digits (CVD) 325
12.2.2 Transactions Where CVD Is Required 325
12.2.3 CVD Information Object 326
12.2.4 CVD Result Codes 326

Appendix A Definition of Request Fields 328
A.1 Definition of Request Fields – Admin Transactions 333
A.2 Definition of Request Fields – Vault 334
A.3 Definition of Request Fields – 3-D Secure 2.2 335
A.4 Definition of Request Fields – Information Objects 342
A.5 Definition of Request Fields – Credential on File 343
A.6 Definition of Request Fields – Installments by Visa 345
A.7 Definition of Request Fields – Apple Pay Token 346
A.8 Definition of Request Fields – GooglePay Token Temp Add 348
A.9 Definition of Request Fields – Recurring Billing 349
A.10 Definition of Request Fields – Account Name Verification Object 350
A.11 Definition of Request Fields – AVS Info Object 350
A.12 Definition of Request Fields – CVD Info Object 351

Appendix B Definitions of Response Fields 352
B.1 Definition of Response Fields –Installments by Visa 370

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 specialist and need technical development assistance, contact our eProducts Technical Consultants: 1-866-319-7450 api@moneris.com	If your application is already live and you need production support, contact Moneris Customer Service: onlinepayments@moneris.com 1-866-319-7450 Available 24/7

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

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

Changes in v1.6.9

- Added Vault Tokenize Credit Card request to the Vault transaction set. This transaction allows for tokenizing a card used in a previous financial transaction without resubmitting card data
- Added `return_issuer_id` to the Definition of Request Fields for Vault.
- Added new values to `request_type` in 3DS requests.
- Updated Visa Secure (3DS) support for the field ri_indicator to support 01, 02, 06, 07, and 11 for Payment Transactions and 03, 04, 05, and 10 for Non-Payment Transactions

Changes in v1.6.8

- Added browser_ip_address to MPI 3DS Authentication Request - Browser Channel
- Added [[[Undefined variable Global-Request-Fields/Global-3DS-Req-Flds.Cardholder-Phone-Work]]], [[[Undefined variable Global-Request-Fields/Global-3DS-Req-Flds.Cardholder-Phone-Home]]] and [[[Undefined variable Global-Request-Fields/Global-3DS-Req-Flds.Cardholder-Phone-Mobile]]] to MPI 3DS Authentication Request - Browser Channel
- Added new request object and fields to support Visa Account Name Verification as an option within the basic Card Verification transaction. See `account_name_verification` and its subfields `first_name, middle_name, and last_name`
- Added response field `AccountNameVerificationResultCode` to the core response field definitions as part of the new Visa Account Name Verification

Changes in v1.6.6

- Added 3 new transaction types GooglePay Token Temp Add, GooglePay Token Purchase, and GooglePay Token Preauth. These transactions allow for sending an encrypted GooglePay payload and receiving a Moneris temporary token in exchange for processing 3DS authentication.
- Added new request objects and fields to support GooglePay Token transactions such as `PaymentToken` and its subfields `signature, protocol version, and signed message`
- Added new response field `GooglePayPaymentMethod`

- Updated email addresses for development support toapi@moneris.com
- Updated Server to Server endpoints
- Updated date format for recurring start date to YYY/MM/DD

Changes in v1.6.5

- Removed the request_type field from MPI 3DS Authentication Request - 3RI with recurring
- Updated the note for the fields recurring_frequency and recurring_expiry
- Removed three_ds_completion_ind in MPI 3DS Authentication Request - 3RI with recurring/no recurring
- Moved prior_request_auth_info field to optional section for MPI 3DS Prior Authentication Info for MPI 3DS Authentication Request - 3RI with recurring
- Updated the note related to the Field email
- Added Merchant Advice Code field in the Appendix B - Definitions of Response Fields

Changes in v1.6.4

- Updated the 3-D version to 2.2
- Added the 3RI flow chart to the section the section “Transaction Flow for 3-D Secure - 3RI channel”
- Updated the field Email moving from optional to required
- Added a note to the field Email
- Added the status “D” in the TransStatus Code
- Updated TransStatusReason Decline Codes

- Removed the Visa Checkout section. Visa Checkout has been decommissioned in June 1st 2023
- Updated the note related to the Field DS transaction ID
- Updated the note related to the Field 3DS Version
- Updated the note related to the Field 3DS server transaction ID
- Added a comment to Visa Secure
- Updated the note related to the Field ri_indicator

Changes in v1.6.3

- Added Pay By Bank chapter with topics covering the new feature, the Konek authentication flow, and how to build and test an integration
- Added Pay By Bank optional object to Purchase and Preauthorization basic transactions. This object contains new fields for linking a financial transaction to a Pay By Bank authentication, such as consent_id, cryptogram, cryptogram_expiry, payment_method
- Added Pay By Bank transactions Get Access Token, Get Consent Data, and Get Payment Transaction Data
- Added the Pay By Bank response objects and their fields to the Definition of Response Fields in the Appendix. Each object has its own sub-topic to avoid confusion.
- Added the Pay By Bank request fields to the Definition of Request Fields in the Appendix.
- Added PBB to the list of possible values for wallet_indicator

Changes in v1.6.2

- Added note to ds_trans_id on only submitting it in financial transactions if using a 3rd party 3DS Secure service

Changes in v1.6.1

- Added new 3DS fields to 3DS Authentication to support 3RI such as message_category, device_ channel, ri_indicator
- Added new 3DS fields to 3DS Authentication to support 3RI Decoupled Authentication such as decoupled_request_indicator, decoupled_request_max_time, decoupled_request_async_url
- Added new 3DS object prior_request_auth_data to 3DS Authentication to support 3RI, including its fields prior_request_auth_info , prior_request_auth_method, prior_request_auth_ref, prior_ request_auth_timestamp
- Added additional fields to 3DS responses threeds_version and AuthenticationType
- Added additional topics on 3DS Authentication for the 3RI scenarios with and without recurring features
- Retitled the existing 3DS Authentication scenario to specify it is intended for the browser channel only
- Added topic on 3RI channel authentication flow and retitled previous 3DS flow to specify it is intended for browser channel only
- Added topic on Handling 3RI Decoupled Authentication flow to explain the asynchronous response handling
- Added topic on Server To Server endpoints to cover the separate URL for 3DS Authentication

Changes in v1.6.0

- Added new foreign_indicator field to Basic Transaction set: purchase and preauth
- Added new foreign_indicator field to 3-D Secure Transaction set: cavvPurchase and cavvPreauth
- Added new foreign_indicator field to the Basic Request DTD
- Added new foreign_indicator values in Appendix A Definition of Request Fields
- Updated AVS Response Codes table
- Added InstallmentResults to the Basic Response DTD and Vault Response DTD

- Added Installment Info Object to Purchase, Pre-Authorization, resPurchaseCC and resPreauthCC in the Basic Request DTD
- Added Installment Info Object to resPurchaseCC and resPreauthCC in the Vault Request DTD
- Added resInstallmentLookup to the Vault Request DTD
- Added Elements returned with resInstallmentLookup transactions to the Vault Response DTD
- Added new section and topics about Installments by Visa
- Added topic about Vault and Installments in the Vault section
- Added Installment Info Object to Purchase, Pre-Authorization, resPurchaseCC and resPreauthCC
- Added Installments by Visa transactions to the Basic Request DTD
- Added Installments by Visa fields to the Basic Response DTD

Changes in v1.4.3

- Added Increment Preauthorization in Basic transactions
- Added request field is_incremental in Basic, Vault, and 3-D Secure preauthorization transactions

Changes in v1.4.3

- Added section for Google Pay™ transactions
- Added request field **DS transaction ID** in

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>`).

1.2 Server to Server Endpoints

XML transactions sent to the Moneris Gateway use two endpoints depending on whether you are utilizing our Moneris 3DS Authentication Server or

Testing URLs:

Default endpoint:

https://mpg1t.moneris.io/gateway2/servlet/MpgRequest

MPI (3DS card lookup, authentication, and CAVV lookup:

https://mpg1t.moneris.io/mpi2/servlet/MpiServlet

Production URLs:

Default endpoint:

https://mpg1.moneris.io/gateway2/servlet/MpgRequest

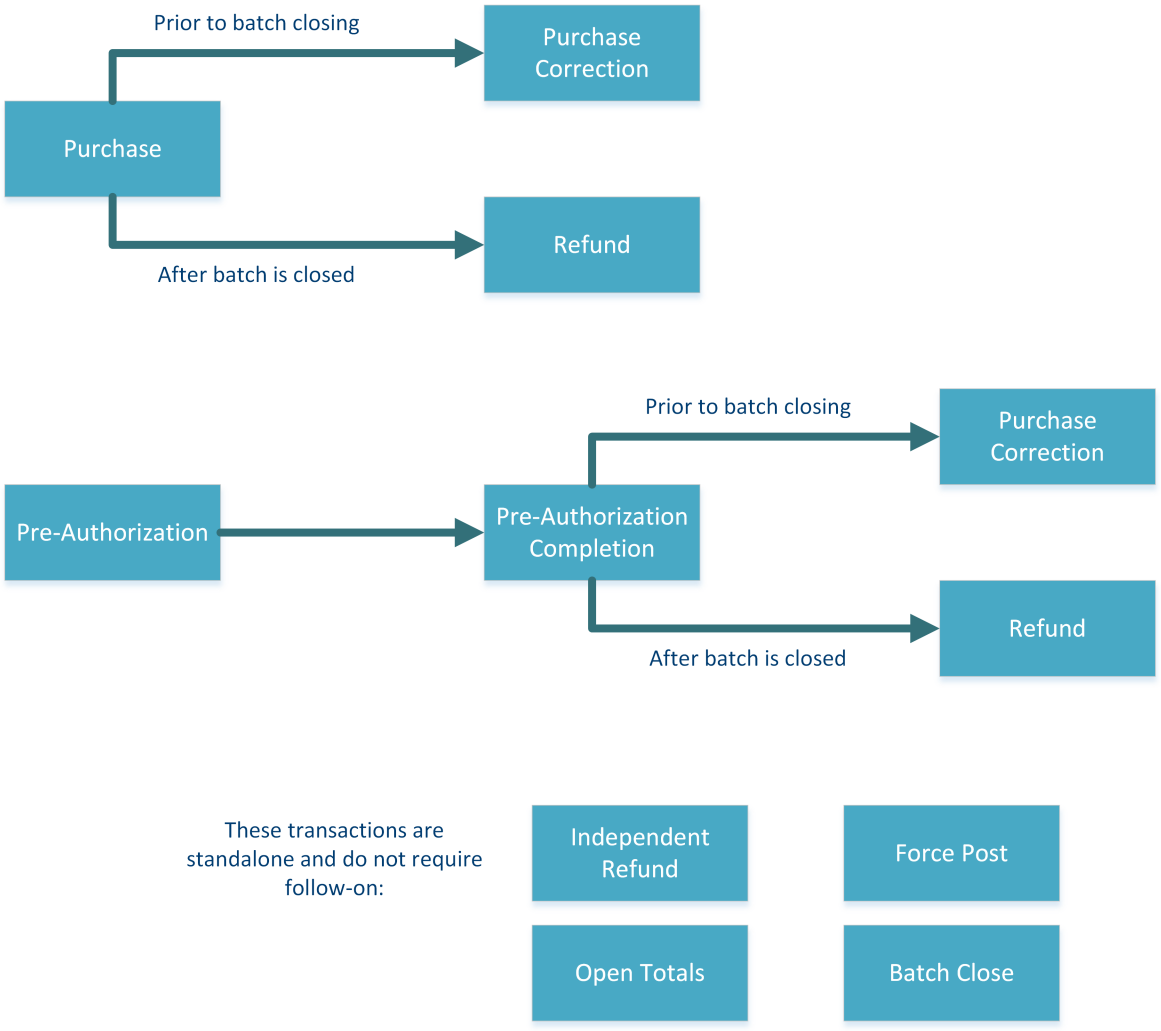
MPI (3DS card lookup, authentication, and CAVV lookup:

https://mpg1.moneris.io/mpi2/servlet/MpiServlet

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
- 2.7 Re-Authorization
- 2.8 Pre-Authorization Completion
- 2.9 Force Post
- 2.10 Refund
- 2.11 Independent Refund
- 2.12 Purchase Correction
- 2.13 Card Verification
- 2.14 Batch Close
- 2.15 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 | mcp_cavv_preauth | mcp_cavv_purchase | mcp_cavv_res_preauth_cc |
mcp_cavv_res_purchase_cc | res_cavv_preauth_cc | res_cavv_purchase_cc | res_add_cc | res_
update_cc | res_delete | batchclose | opentotals | recur_update | applepay_token_purchase |
applepay_token_preauth | googlepay_purchase | googlepay_preauth | installmentLookup |
resinstallmentLookup))>

<!ELEMENT store_id (#PCDATA)>
<!ELEMENT api_token (#PCDATA)>
<!ELEMENT status_check (#PCDATA)>
```

```
<!--The following are the basic credit card transactions -->

<!ELEMENT purchase (order_id, cust_id?, amount, pan, expdate, crypt_type, dynamic_
descriptor?, cust_info?, avs_info?, cvd_info?, pbb_info?, recur?,cof_info?, installment_
info?, wallet_indicator?, foreign_indicator?)>
<!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?, pbb_info?, installment_info?, wallet_indicator?,
foreign_indicator?)>
<!ELEMENT incremental_preauth (order_id, txn_number, amount)>
<!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?, account_name_verification?)>
<!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. -->
<!--NOTE: In Frictionless flow, you may receive TransStatus as "Y", in which case you can
then proceed directly to Cavv Purchase/Preauth with values below -->

<!--NOTE: threeds_version and threeds_server_trans_id are mandatory for 3DS Version 2.2+ -->

<!ELEMENT cavv_preauth (order_id , cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, cust_info?, avs_info?, cvd_info?, cof_info?, ds_
trans_id?, foreign_indicator?)>
<!ELEMENT cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, cust_info?, avs_info?, cvd_info?, recur?, cof_info?,
ds_trans_id?, foreign_indicator?)>

<!--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)>
<!--NOTE: threeds_version and threeds_server_trans_id are mandatory for 3DS Version 2.0+ -->
<!ELEMENT mcp_cavv_preauth (order_id , cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_server_trans_id, cust_info?,
avs_info?, cvd_info?, cof_info?, ds_trans_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
```

```
<!ELEMENT mcp_cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_server_trans_id, cust_info?,
avs_info?, cvd_info?, recur?, cof_info?, ds_trans_id?, mcp_version, cardholder_amount,
cardholder_currency_code, mcp_rate_token?)>
<!ELEMENT mcp_cavv_res_preauth_cc (data_key, order_id, cust_id, crypt_type, dynamic_
descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_
token?, threeds_version, threeds_server_trans_id, ds_trans_id?)>
<!ELEMENT mcp_cavv_res_purchase_cc (data_key, order_id, cust_id, crypt_type, dynamic_
descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_
token?, threeds_version, threeds_server_trans_id, ds_trans_id?)>

<!--The following are the Vault transactions -->
<!--NOTE: threeds_version and threeds_server_trans_id are mandatory for 3DS Version 2.0+ -->
<!ELEMENT res_cavv_preauth_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, threeds_version, threeds_server_trans_id, ds_trans_id?)>
<!ELEMENT res_cavv_purchase_cc (data_key, order_id, cust_id, crypt_type, dynamic_descriptor?,
expdate?, threeds_version, threeds_server_trans_id, ds_trans_id?)>
<!ELEMENT res_add_cc (pan, expdate, crypt_type, cust_id, phone, email, note, data_key_
format?)>
<!ELEMENT res_update_cc (data_key, cust_id, phone, email, note, pan, expdate, crypt_type)>
<!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?, installment_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?, installment_info?)>
<!ELEMENT res_ind_refund_cc (data_key, order_id, cust_id?, amount, crypt_type, cof_info?)>
<!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 res_mpitxn (data_key, xid, amount, MD, merchantUrl, accept, userAgent, expdate?)>
<!--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_recurs?,
total_num_recurs?, 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 2.2 transactions -->
<!ELEMENT Mpi2Request (store_id, api_token, (card_lookup | threeds_authentication | cavv_
lookup))>

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

```
//Browser Channel only
<!ELEMENT threeds_authentication (message_category, device_channel, request_type, order_id,
(pan | data_key, expdate), amount, currency?, cardholder_name, threeds_completion_ind, bill_
address1, bill_province, bill_city, bill_postal_code, bill_country, ship_address1, ship_
province, ship_city, ship_postal_code, ship_country, notification_url, challenge_windowsize,
browser_useragent, browser_java_enabled, browser_screen_height, browser_screenwidth, browser_
language, email?, request_challenge?)>

//3RI, non-recurring
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_request_
indicator?, decoupled_request_max_time?, decoupled_request_async_url?, ri_indicator, prior_
authentication_info?, order_id, (pan | data_key, expdate), amount, currency?, cardholder_
name), bill_address1, bill_province, bill_city, bill_postal_code, bill_country, ship_
address1, ship_province, ship_city, ship_postal_code, ship_country,>

//3RI, recurring
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_request_
indicator?, decoupled_request_max_time?, decoupled_request_async_url?, recurring_frequency,
recurring_expiry, ri_indicator, prior_authentication_info, order_id, (pan | data_key,
expdate), amount, currency?, cardholder_name), bill_address1, bill_province, bill_city, bill_
postal_code, bill_country, ship_address1, ship_province, ship_city, ship_postal_code, ship_
country,>

<!ELEMENT prior_authentication_info (prior_request_auth_data, prior_request_ref, prior_
request_auth_method>)

<!ELEMENT cavv_lookup (cres)>

<!--The following are the Installments by Visa transactions -->

<!ELEMENT installmentLookup (store_id, api_token, order_id, amount, pan, expdate)
<!ELEMENT resInstallmentLookup (store_id, api_token, order_id, amount, data_key, expdate)

<!-- 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_recurs (#PCDATA)>
<!ELEMENT total_num_recurs (#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 Installment Info -->
<!ELEMENT installment_info (plan_id, plan_id_ref, tac_version)>

<!-- 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_recurs, period, recur_amount)>
<!ELEMENT recur_unit (#PCDATA)>
<!ELEMENT start_now (#PCDATA)>
<!ELEMENT start_date (#PCDATA)>
<!ELEMENT num_recurs (#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 PBB -->

<!ELEMENT pbb_info (consent_id, payment_method, cryptogram?, cryptogram_expiry?)>
<!ELEMENT consent_id (#PCDATA)>
<!ELEMENT payment_method (#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 payment_token (signature, protocol_version, signed_message)>
<!ELEMENT displayName (#PCDATA)>
<!ELEMENT network (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT header (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT protocol_version (#PCDATA)>
<!ELEMENT signed_message (#PCDATA)>

<!-- start 3DS 2.2 specific fields -->
<!ELEMENT threeds_version (#PCDATA)>
<!ELEMENT threeds_server_trans_id (#PCDATA)>
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT notification_url (#PCDATA)>
<!ELEMENT cardholder_name (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT threeds_completion_ind (#PCDATA)>
<!ELEMENT request_type (#PCDATA)>
<!ELEMENT purchase_date (#PCDATA)>
<!ELEMENT challenge_windowsize (#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)>
<!ELEMENT message_category (#PCDATA)>
<!ELEMENT device_channel (#PCDATA)>
<!ELEMENT decoupled_request_indicator (#PCDATA)>
<!ELEMENT decoupled_request_max_time (#PCDATA)>
<!ELEMENT decoupled_request_async_url (#PCDATA)>
<!ELEMENT recurring_frequency (#PCDATA)>
<!ELEMENT recurring_expiry (#PCDATA)>
<!ELEMENT ri_indicator (#PCDATA)>
<!ELEMENT prior_request_auth_data (#PCDATA)>
<!ELEMENT prior_request_ref (#PCDATA)>
<!ELEMENT prior_request_auth_method (#PCDATA)>
<!ELEMENT prior_request_auth_timestamp (#PCDATA)>

<!-- start Vault specific fields -->
```

```
<!ELEMENT phone (#PCDATA)>
<!ELEMENT note (#PCDATA)>
<!ELEMENT data_key_format (#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, IssuerId?,
MCPRateToken?, RateTxnType?, (Rate+)?, RateInqStartTime?, RateInqEndTime?,
RateValidityStartTime?, RateValidityEndTime?, RateValidityPeriod?,
CardholderCurrencyCode?, CardholderAmount?, MerchantSettlementCurrency?,
MerchantSettlementAmount?, MCPRate?, MCPErrorStatusCode?, MCPErrorMessage?, ResolveData?,
MpiType?, MpiSuccess?, MpiMessage?, MpiPaReq?, MpiTermUrl?, MpiMD?, MpiACSUrl?, MpiCavv?,
MpiPAResVerified?) |
(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 2.2 transactions -->
<!ELEMENT MessageType (#PCDATA)>
<!ELEMENT ThreeDSMethodURL (#PCDATA)>
<!ELEMENT ThreeDSMethodData (#PCDATA)>
<!ELEMENT ChallengeURL (#PCDATA)>
<!ELEMENT ChallengeData (#PCDATA)>
<!ELEMENT ChallengeCompletionIndicator (#PCDATA)>
<!ELEMENT TransStatus (#PCDATA)>
<!ELEMENT ThreeDSServerTransId (#PCDATA)>
<!ELEMENT ECI (#PCDATA)>
<!ELEMENT Cavv (#PCDATA)>
<!-- The following are only applicable to Vault (res_) transactions -->
<!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?)>
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT payment_type (#PCDATA)>
<!ELEMENT cust_id (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT note (#PCDATA)>
<!ELEMENT pan (#PCDATA)>
<!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_mpitxn -->
<!ELEMENT MpiType (#PCDATA)>
<!ELEMENT MpiSuccess (#PCDATA)>
<!ELEMENT MpiMessage (#PCDATA)>
<!ELEMENT MpiPaReq (#PCDATA)>
<!ELEMENT MpiTermUrl (#PCDATA)>
<!ELEMENT MpiMD (#PCDATA)>
<!ELEMENT MpiACSUrl (#PCDATA)>
<!ELEMENT MpiCavv (#PCDATA)>
<!ELEMENT MpiPAResVerified (#PCDATA)>
<!-- the following are only returned with installmentLookup and resInstallmentLookup
transactions -->
<!ELEMENT EligibleInstallmentPlans (PlanCount, PlanDetails (PlanId, PlanIdRef, Name, Type,
NumInstallments, InstallmentFrequency, TotalFees, TotalPlanCost, APR, Tac (TacCount,
TacDetails (Text, Url, Version, LanguageCode)), PromotionInfo (PromotionCode, PromotionId),
FirstInstallment (UpfrontFee, Amount), LastInstallment (InstallmentFee, Amount))>
```

```
<!-- the following are only returned in the response for Visa Installments with Purchase,
Preauth, res_Purchase and res_Preauth transactions -->
<!ELEMENT InstallmentResults (PlanID, PlanRef, TacVersion, PlanAcceptanceID, PlanStatus,
PlanResponse)>
```

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?, PBB_
info?, installment_info?, wallet_indicator?, foreign_indicator?)>
```

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>	*Boolean* 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

Purchase transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
credit card number	*String*	Credit card number, usually 16 digits

Variable Name	Type and Limits	Description
<pan >	max 20-character alpha-numeric	—field can be maximum 20 digits in support of future expansion of card number ranges. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
electronic commerce indic-ator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allow-able 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-action (3-D Secure) 7 – SSL-enabled merchant **In Credential on File transactions where the request field e-commerce indicator is also being sent**: the allow-able 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 = V, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values

Variable Name	Type and Limits	Description
		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 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 **NOTE:** The 22-character maximum limit must take the "/" into account as one of the characters
foreign indicator <foreign_indicator>	*Boolean* true or false	Used to identify domestic transactions processed by a marketplace merchant that is in a different country.
wallet indicator <wallet_indicator>	*String* 3-character alphanumeric	Indicates when a card number has

Variable Name	Type and Limits	Description
		been collected via a digital wallet, such as in Apple Pay, Google Pay™, Visa Checkout and Mastercard MasterPass, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
Customer Information <cust_info> For information on request fields for this object, see "Customer Information Object" on page 315	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information	*Object*	Contains fields applying to the Address Verification Service (AVS) e-

Variable Name	Type and Limits	Description
<avs_info> For information on request fields for this object, see "Definition of Request Fields – AVS Info Object" on page 350	N/A	fraud tool
CVD Information <cvd_info> For information on request fields for this object, see "Definition of Request Fields – CVD Info Object" on page 351	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur> For information on request fields for this object, see "Definition of Request Fields – Recurring Billing" on page 349	*Object* N/A	Contains fields related to Recurring Billing
Credential on File Information <cof_info> For information on request fields for this object, see "Definition of Request Fields – Credential on File" on page 343	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Pay By Bank Info <PBB_info> For information on request fields for this object, see "Definition of Request Fields – PBB Object" on page 1	*Object* N/A	Required when performing a financial transaction after obtaining an Interac Konek (Pay By Bank) authentication. Links the PBB consent to this transaction.

Variable Name	Type and Limits	Description
Installment Info For fields in this object, see 6.6 Installment Info Object	Object N/A	Contains request fields related to installments

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?, installment_
info?, wallet indicator?, foreign_indicator?, is_incremental?)>
```

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/

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>	*Boolean* 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

Pre-Authorization transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*String* 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal	Transaction dollar amount This must contain at least 3 digits, two of which are penny values Minimum allowable value = $0.01, maximum allowable value =

Variable Name	Type and Limits	Description
	point **EXAMPLE:** 1234567.89	$9999999.99
credit card number <pan>	*String* 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. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
electronic commerce indic-ator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allow-able 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-action (3-D Secure) 7 – SSL-enabled merchant **In Credential on File transactions where the request field e-commerce indicator is also being sent**: the allow-able values for e-commerce indicator are dependent on the value sent for **payment indicator**, as follows:

Variable Name	Type and Limits	Description
		if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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

Pre-Authorization 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 **NOTE:** The 22-character maximum limit must take the "/" into account as one of the characters
foreign indicator	*Boolean*	Used to identify domestic transactions

Variable Name	Type and Limits	Description
<foreign_indicator>	true or false	processed by a marketplace merchant that is in a different country.
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
is incremental is_incremental	*Boolean* true/false	Indicates if this preauthorization is using an estimated amount. Estimations allow for incrementing the amount held via subsequent incrementalAuth requests. Defaults to

Variable Name	Type and Limits	Description
		false. **NOTE:** Please note that if this field is true, the preauthorization is only eligible for a single Preauthorization Completion. Any completion sent for partial completion is treated as a full completion (ship_indicator= P is treated as = F when is_incremental= true on the original preauth)
Customer Information <cust_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields related to Recurring Billing
Credential on File Information	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

Variable Name	Type and Limits	Description
<cof_info> For information on request fields for this object, see xref-fHere		
Installment Info For fields in this object, see 6.6 Installment Info Object	Object N/A	Contains request fields related to installments

2.6 Incremental Pre-Authorization

Increases the locked amount of funds in an existing pre-authorization for later settle by a single pre-authorization completion. There is no limit to the number of incremental pre-authorization transactions on the original estimated auth and each new incremental pre-authorization increases the hold on the customer's credit card.

Incremental Pre-authorizations require an estimated amount in the initial Pre-Authorization. This is set using the <is_incremental> field set to true.

For Mastercard only, an Incremental Pre-Authorization can be submitted with a $0 value for the amount to request extending the allowable timeframe for completion (e.g, 30 days).

For additional details on using estimated amounts in Pre-Authorizations and using Incremental Pre-Authorizations to increase the locked amount of funds, see 1 Incremental Authorization Rules

XML transaction object

<incremental_preauth>

Increment Pre-Authorization transaction object definition

```
<!ELEMENT incremental_preauth (order_id, txn_number, amount)>
```

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
		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>	*Boolean* 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

Increment Pre-Authorization transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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.

Variable Name	Type and Limits	Description
transaction number <txn_number>	*String* 255-character, alpha-numeric, hyphens or under-scores variable length	Used to reference the original trans-action 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: ref-erences a Pre-Authorization Refund/Purchase Correction: ref-erences a Purchase or Pre-Author-ization Completion
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 to increase the preauthorization by. This must contain at least 3 digits, two of which are penny values Minimum allowable value = $0.01, maximum allowable value = $9999999.99

2.7 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	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>	*Boolean* 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

TransactionTopicName transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction iden-

Variable Name	Type and Limits	Description
<xmlvariableHere>	50-character alphanumerica-Z A-Z 0-9 _ - : . @ spaces	tifier 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.
original order ID <orig_order_id>	*String* 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
transaction number <txn_number>	*String* 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 Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion
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
electronic commerce indic-	*String*	Describes the category of e-com-

Variable Name	Type and Limits	Description
ator <crypt_type>	1-character alphanumeric	merce 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 = V, 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

TransactionTopicName transaction request fields – Optional

Variable Name	Type and Limits	Description
customer ID	*String*	Merchant-defined field that can be used as an identifier

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

2.8 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
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>	*Boolean* 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

Pre-Authorization Completion transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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.
completion amountamount <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
transaction number	*String*	Used to reference the original trans-

Variable Name	Type and Limits	Description
<txn_number>	255-character, alpha-numeric, hyphens or under-scores variable length	action 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>	*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 = V, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = C, then allowable values

Variable Name	Type and Limits	Description
		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

Pre-Authorization Completion transaction request fields – Optional

Variable Name	Type and Limits	Description
Pay By Bank Info <PBB_info> For information on request fields for this object, see "Definition of Request Fields – PBB Object" on page 1	*Object* N/A	Required performing a completion transaction for an Account Based Payment (ABP) for Pay By Bank. Utilize Get Transaction Data to obtain the cryptogram. Links the PBB consent to this transaction.
shipping indicator <ship_indicator>	*String* 1-character alphanumeric	Used to identify completion transactions that require multiple shipments, also referred to as multiple completions By default, if shipping indicator is not sent, the Pre-Authorization Completion is listed as final To indicate that the Pre-Authorization Completion is to be left open by the issuer as supplemental shipments or completions are pending, submit shipping indicator with a value of P Possible values: P – Partial F – Final

2.9 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	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>	*Boolean* 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

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

Force Post transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
credit card number <pan>	*String* 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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 4-character alphanumeric	Expiry date of the credit card, in YYMM format.

Variable Name	Type and Limits	Description
	YYMM	**NOTE:** This is the reverse of the MMYY date format that is presented on the card.
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 = V, 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>	*String* 8-character alphanumeric	An authorization code required to carry out a Force Post; provided in the transaction response from the issuing

Variable Name	Type and Limits	Description
		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 **NOTE:** The 22-character maximum limit must take the "/" into account as one of the characters

2.10 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	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>	*Boolean* 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

Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
transaction number <txn_number>	*String* 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>	*String* 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are:

Variable Name	Type and Limits	Description
		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 = V, 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>	*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

Variable Name	Type and Limits	Description
Pay By Bank Info <PBB_info> For information on request fields for this object, see "Definition of Request Fields – PBB Object" on page 1	*Object* N/A	Required performing a refund transaction for an Account Based Payment (ABP) for Pay By Bank. Utilize Get Transaction Data to obtain the cryptogram. Links the PBB consent to this transaction.
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

2.11 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	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>	*Boolean* 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 transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alpha-numerica-Z A-Z 0-9 _ - : . @ spaces	Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
credit card number <pan>	*String* 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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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

Variable Name	Type and Limits	Description
		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 = V, 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

Independent Refund 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	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement

Variable Name	Type and Limits	Description
	total of 22 characters including your merchant name and separator **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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

2.12 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.

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	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

Variable Name	Type and Limits	Description
		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>	*Boolean* 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

Purchase Correction transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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	*String*	Used to reference the original transaction when performing a follow-on

Variable Name	Type and Limits	Description
<txn_number>	255-character, alpha-numeric, hyphens or under-scores variable length	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>	*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 = V, 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
		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 Correction transaction request fields – Optional

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

2.13 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?, account_name_verification?)>
```

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>	*Boolean* 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

Card Verification transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alpha-numerica-Z A-Z 0-9 _ - : . @ spaces	Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
credit card number <pan>	*String* 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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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**

Variable Name	Type and Limits	Description
		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 = V, 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

Card Verification 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
AVS Information <avs_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> For information on request	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool

Variable Name	Type and Limits	Description
fields for this object, see xre-fHere		
Credential on File Information <cof_info> For information on request fields for this object, see xre-fHere	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Visa Account Name Verification Object account_name_verification For information on request fields for this object, see "Definition of Request Fields – Account Name Verification Object" on page 350	*Object* N/A	Contains cardholder account name for verification. Only applicable to Visa credit cards.

2.14 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

Variable Name	Type and Limits	Description
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>	*Boolean* 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>	*String* N/A	Identification number assigned to a particular electronic cash register; provided by Moneris

2.15 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>	*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>	*Boolean* 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

Open Totals transaction request fields – Required

Variable Name	Type and Limits	Description
electronic cash register (ECR) number <ecr_number>	*String* 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 A.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-intiated 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 4.5.9 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 2.13 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

0.1 Definition of Request Fields – Credential on File

Variable Name	Type and Limits	Description
issuer ID <issuer_id>	*String* 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
		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) 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 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
payment indicator <payment_indicator>	*String* 1-character alphabetic	Indicates the current or intended use of the credentials Possible values for first transactions: C - unscheduled Credential on File (first transactions only) R - recurring V - recurring variable payment transaction Possible values for subsequent transactions: R - recurring V - recurring variable payment transaction U - unscheduled merchant-initiated transaction Z - unscheduled customer-initiated transaction 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:

Variable Name	Type and Limits	Description
		if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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
payment information <payment_information>	*String* 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 and Installments
- 4.5 Vault Administrative Transactions
- 4.6 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 | resinstallmentLookup)>
<!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?, installment_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?, installment_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)>
<!--The following are the Installments by Visa transactions -->
<!ELEMENT resInstallmentLookup (store_id, api_token, order_id, amount, data_key, expdate)

<!-- 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_recurs, period, recur_amount)>
<!ELEMENT recur_unit (#PCDATA)>
<!ELEMENT start_now (#PCDATA)>
<!ELEMENT start_date (#PCDATA)>
<!ELEMENT num_recurs (#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)>
<!-- start installment info -->
<!ELEMENT installment_info (plan_id, plan_id_ref, tac_version)>
```

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)>

<!-- the following are only returned with resInstallmentLookup transactions -->
<!ELEMENT EligibleInstallmentPlans (PlanCount, PlanDetails (PlanId, PlanIdRef, Name, Type,
NumInstallments, InstallmentFrequency, TotalFees, TotalPlanCost, APR, Tac (TacCount,
TacDetails (Text, Url, Version, LanguageCode)), PromotionInfo (PromotionCode, PromotionId),
FirstInstallment (UpfrontFee, Amount), LastInstallment (InstallmentFee, Amount))>

<!-- the following are only returned in the response for Visa Installments with res_Purchase
and res_Preauth transactions -->
<!ELEMENT InstallmentResults (PlanID, PlanRef, TacVersion, PlanAcceptanceID, PlanStatus,
PlanResponse)>
```

4.4 Vault and Installments

Installments functionality is also available on transactions using cardholder credentials stored in the Moneris Vault. To offer this feature to the customer, send the Vault Installment Plan Lookup transaction prior to running a Purchase with Vault or Pre-Authorization with Vault.

For more about Installments, see 6 Installments by Visa

4.5 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.

4.5.1 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	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>	*Boolean* 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>	*String* 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. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
electronic commerce indic-ator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allow-able 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-action (3-D Secure) 7 – SSL-enabled merchant **In Credential on File transactions where the request field e-commerce indicator is also being sent**: the allow-able 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 = V, then allowable values

Variable Name	Type and Limits	Description
		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

Vault Add Credit Card 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
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	*String* 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	*String* 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 <avs_info>	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool

4.5.2 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

Variable Name	Type and Limits	Description
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>	*Boolean* 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 Update Credit Card transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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

Vault Update Credit Card 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
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	*String* 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	*String* 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>	*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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
electronic commerce indicator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-commerce transaction being processed. Allowable values are:

Variable Name	Type and Limits	Description
		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 = V, 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
AVS Information <avs_info> For information on request fields for this object, see xrefHere	*String* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
Credential on File Information <cof_info>	*String* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

Variable Name	Type and Limits	Description
For information on request fields for this object, see xref Here		

4.5.3 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>	*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>	*Boolean* 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 Delete transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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

4.5.4 Vault Tokenize Credit Card - res_tokenize_cc

Creates a new credit card profile using the credit card number, expiry date and e-commerce indicator that were submitted in a previous financial transaction. Previous transactions to be tokenized must have included the Credential on File Info object.

The Issuer ID received in the previous transaction response is sent in the Vault Tokenize Credit Card request to reference that this is a stored credential. If you require this Issuer ID in the response to this request, include **return_issuer_ID** as `true`; this allows for retrieval of the Issuer ID from the previous financial transaction.

Basic transactions that can be tokenized are:

- Purchase
- Pre-Authorization
- Card Verification

The tokenization process is outlined below :

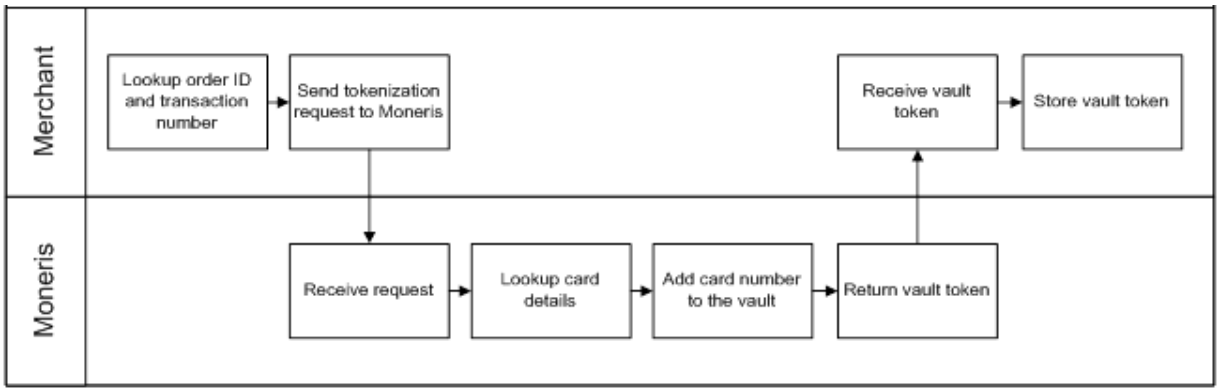


Figure 1: Tokenize process diagram

[[[Undefined variable Heading.Level1]]] transaction object definition

```
<!ELEMENT res_token_cc (order_id, txn_number, return_issuer_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	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>	*Boolean* 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

[[[Undefined variable Heading.Level1]]] 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.
transaction number <txn_number>	*String* 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

[[[Undefined variable Heading.Level1]]] transaction request fields – Optional

Variable Name	Type and Limits	Description
return issuer ID <return_issuer_id>	*Boolean* true/false	When true, Gateway returns the bank Issuer ID. Defaults to False.
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
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	*String* 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	*String* 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 <avs_info>	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool For more information about AVS, see 12.1.2 AVS Information Object

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

4.5.5 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>	*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
		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>	*Boolean* 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 Lookup Full transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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

4.5.6 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>	*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>	*Boolean* true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with

Variable Name	Type and Limits	Description
		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 Lookup Masked transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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

4.5.7 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.

NOTE: This transaction supports both temporary and permanent tokens.

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>	*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>	*Boolean* 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 Is Corporate Card 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

4.5.8 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	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your test or production store's Admin set-

Variable Name	Type and Limits	Description
		tings 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>	*Boolean* 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 Get Expiring transaction request fields – Required

Vault Get Expiring does not require any transaction request variables.

4.5.9 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	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>	*Boolean* 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>	*String* 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. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
electronic commerce indic-ator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-com-merce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Instal-ment 4 – Mail Order / Telephone Order—Unknown classification 5 – Authenticated e-commerce transaction (3-D Secure) 6 – Non-authenticated e-commerce trans-action (3-D Secure) 7 – SSL-enabled merchant **In Credential on File transactions where the request field e-com-merce 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 val-

Variable Name	Type and Limits	Description
		ues for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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
duration <duration>	*String* 3-character numeric maximum 900 seconds	Amount of time the temporary token should be available

Vault Temporary Token Add transaction request fields – Optional

Variable Name	Type and Limits	Description
data key format <data_key_format>	*String* 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

4.5.10 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.

Using the temporary token, send either a Purchase with Vault, Pre-Authorization with Vault or Card Verification with Vault transaction request including the Credential on File object to get the issuer ID.

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>	*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>	*Boolean* 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 Token – res_add_token transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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>	*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 = V, 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
		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
expiry date <expdate> **NOTE:** This field is optional if you already collected it using the the Hosted Tokenization solution; otherwise, it is required	*String* 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.

Vault Add Token – res_add_token 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
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
email address <email>	*String* 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	*String* 30-character alphanumeric	Used for any supplementary information related to the customer Can be sent in when creating or updating a Vault profile

Variable Name	Type and Limits	Description
AVS Information <avs_info>	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool For more information about AVS, see 12.1.2 AVS Information Object
data key format <data_key_format>	*String* 2-character alphanumeric	Specifies the data key format being returned If left blank, data key format will default to 25-character alpha-numeric Possible values: 0 – 25 character alphanumeric data key 0U – unique 25-character alphanumeric data key

4.6 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.

4.6.1 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 pro-file?	Passed in?	Version used in response
No	No	Customer ID not used in transaction
No	Yes	Passed in
Yes	No	Profile

Already in profile?	Passed in?	Version used in response
Yes	Yes	Passed in

4.6.2 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?, installment_info?, get_
nt_response?)>

<!-- 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	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

Variable Name	Type and Limits	Description
		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>	*Boolean* 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

Purchase with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
order ID <order_id>	*String* 50-character alpha-numerica-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
		have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
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

Variable Name	Type and Limits	Description
		if payment indicator = V, 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>	*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
Customer Information <cust_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> For information on request fields for this object, see xre-	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool

Variable Name	Type and Limits	Description
fHere		
Credential on File Information <cof_info> For information on request fields for this object, see xre-fHere	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Installment Info For fields in this object, see 6.6 Installment Info Object	*Object* N/A	Contains request fields related to installments

4.6.3 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, cavv, crypt_
type, expdate?, threeds_version, threeds_server_trans_id, ds_trans_id?, get_
nt_response?, is_incremental?)>
```

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
		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

s	Type and Limits	Description
status check <status_check>	*Boolean* 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

Pre-Authorization with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alpha-numerica-Z A-Z 0-9 _ - : . @ spaces	Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
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

Variable Name	Type and Limits	Description
		payment indicator, as follows: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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

Pre-Authorization with Vault 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
is incremental is_incremental	*Boolean* true/false	Indicates if this preauthorization is using an estimated amount. Estimations allow for incrementing the amount held via subsequent incrementalAuth requests. Defaults to false. **NOTE:** Please note that if this field is true, the preauthorization is only eligible for a single Preauthorization Completion. Any completion sent for partial completion is treated as a full completion (ship_indicator= P is treated as = F when is_incremental= true on the original preauth)
Customer Information	*Object*	Contains fields that describe miscellaneous customer information,

Variable Name	Type and Limits	Description
<cust_info> For information on request fields for this object, see xref-Here	N/A	billing and shipping information, and item information
AVS Information <avs_info> For information on request fields for this object, see xref-Here	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> For information on request fields for this object, see xref-Here	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool
Credential on File Information <cof_info> For information on request fields for this object, see xref-Here	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Installment Info For fields in this object, see 6.6 Installment Info Object	*Object* N/A	Contains request fields related to installments

4.6.4 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, get_nt_response?)>
```

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>	*Boolean* 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>	*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
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
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

Variable Name	Type and Limits	Description
		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 = V, 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

Independent Refund with Vault 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

4.6.5 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, get_nt_response?)>
```

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>	*Boolean* 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

Force Post with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
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

Variable Name	Type and Limits	Description
		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 = V, 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>	*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 with Vault 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

4.6.6 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?, get_nt_response?)>
```

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>	*Boolean* true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend the original transaction with all the

Variable Name	Type and Limits	Description
		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

Card Verification with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
order ID <order_id>	*String* 50-character alpha-numerica-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.
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

Variable Name	Type and Limits	Description
		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 = V, 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

Card Verification with Vault transaction request fields – Optional

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

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

5 3-D Secure 2.2 Transactions

- 5.1 About 3-D Secure 2.2
- 5.2 Building Your 3-D Secure 2.2 Integration
- 5.3 Implementing Card Lookup Request
- 5.5 Implementing MPI 3DS Authentication Request
- 5.6 Handling the Challenge Flow
- 5.8 Performing the Authorization
- 5.9 Testing Your 3-D Secure 2.2 Integration
- 5.10 Moving to Production With 3-D Secure 2.2
- 5.11 3-D Secure 2.2 TransStatus Codes
- 5.12 3-D Secure 2.2 Commons TransStatusReason Decline Codes
- 5.13 CAVV Result Codes

5.1 About 3-D Secure 2.2

3-D Secure 2.2 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 (please note: Visa Secure does not support all the RI Indicators available in the 3D Secure 2.2. Check the RI Indicators status field to confirm the status Visa Secure support.)
- 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.

The XML 3DS 2.2 API supports two message categories and two device channels from the 3-D Secure authentication protocol:

1. Message Categories:
    - **Payment Authentication** – Cardholder authentication prior to an eCommerce transaction. After a successful 3DS authentication, you proceed with a purchase or pre-authorization.
    - **Non-Payment Authentication (NPA)**– Identity verification and account confirmation performed without an accompanying financial transaction. After a successful 3DS authentication, you might proceed with:
        - Tokenizing the card for future payments
        - Allowing log-in for client portals
        - Any other activity relying on identity or account confirmation

2. Device Channels:
    - **Browser** – The transaction originates from a website utilized via a browser on the cardholder's device.

- For example, an eCommerce transaction originating on the merchant's website with a check-out process that the cardholder is using via their personal computer or mobile phone's web browser (Chrome, Edge, Safari, etc.).

- **3DS Requestor Initiated** – Account confirmations and cardholder authentication with no direct cardholder originating the transaction.
  - 3RI can be used for authenticating Mail-Telephone Order (MOTO) transactions.
  - 3RI can be used to authenticate follow-on transactions as part of a subscription, such as recurring transactions. The first cardholder payment might use a browser-based authentication, with subsequent payments utilizing a 3RI authentication linking to the previous.
  - In situations where a merchant business model accommodates waiting before processing their payment, they can utilize Decoupled Authentication to allow the cardholder to authenticate directly with their issuer via a non-3DS challenge, such as a push notification to a banking app.

5.1.2 Out of Scope/Not Supported Check

- In-app

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 2.0 to 3-D Secure 2.2 Check

The 3DS 2.2 API is different from the 3DS 2.0 API therefore developers will have to complete the steps described in the section 5.2 Building Your 3-D Secure 2.2 Integration.

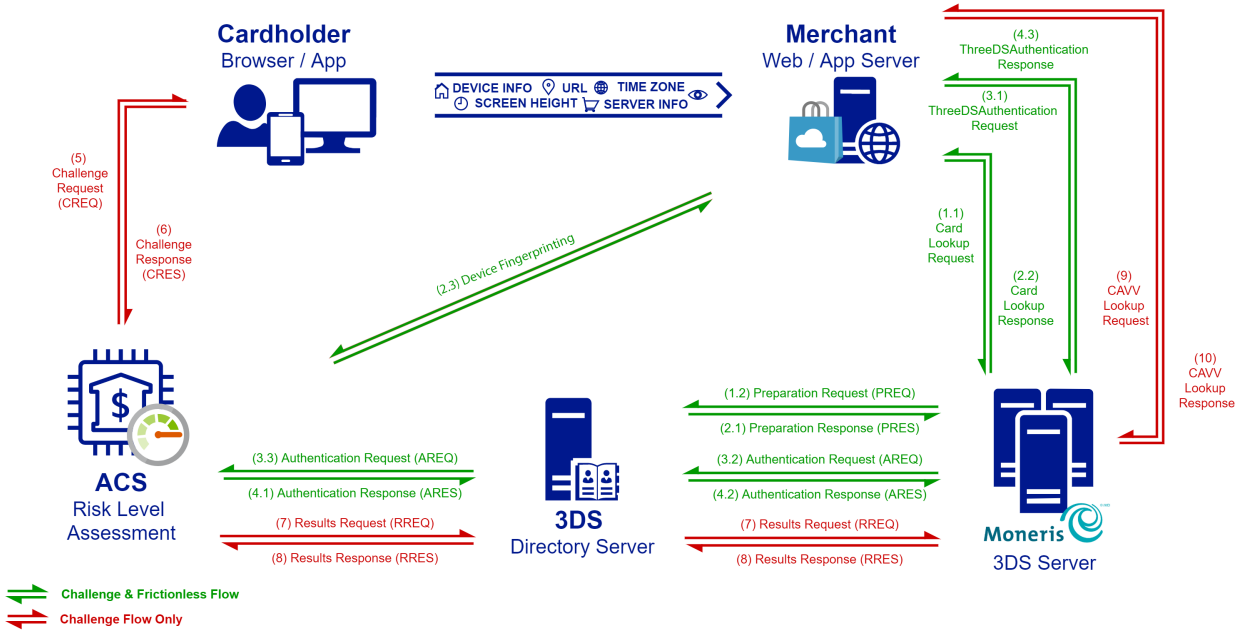
5.2 Building Your 3-D Secure 2.2 Integration

- 5.2.1 Activating 3-D Secure Functionality
- 5.2.2 Transaction Flow for 3-D Secure - Browser channel
- 5.2.3 Transaction Flow for 3-D Secure - 3RI channel

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 - Browser channel



The 3DS 2.2 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 2.2 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 invisible 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.

Steps 1 – 2 (Optional)

An optional card lookup request can be performed to initiate cardholder browser fingerprinting. The merchant website collects device information and provides them to Moneris via the card_lookup request (1.1). Moneris submits this data to the 3DS Directory Server and returns with the card_lookup response containing the card’s supported 3DS version, an ACS URL, and 3DS Method Data representing the fingerprint (2.2). The merchant browser then submits an HTTP POST to the ACS URL with the method data. (2.3)

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 2.2 service so a risk assessment may be initiated.

Steps 3 – 4 (Required)

The 3DS authentication request threeDSAuthentication is performed by the merchant website to initiate validating the cardholder identity. Moneris communicates with the 3DS Directory and the ACS system for that issuer to provide an initial risk assessment (3.2-4.2). Moneris returns a threeDSAuthentication response to the merchant with a TransStatus indicating the action for the website to perform:

- A TransStatus = “Y” or “A” means the website can proceed immediately to the financial transaction with the CAVV value provided. This is a frictionless transaction flow without presenting a challenge.

- A TransStatus = “C” indicates that the cardholder must be presented a challenge. To present the challenge, you must POST a <form> with a “creq” field, which contains the ChallengeData, to the URL defined in the ChallengeURL field.

- A TransStatus = “D” indicates that the cardholder must be presented a challenge via Decoupled Authentication. See Decoupled Authentication.

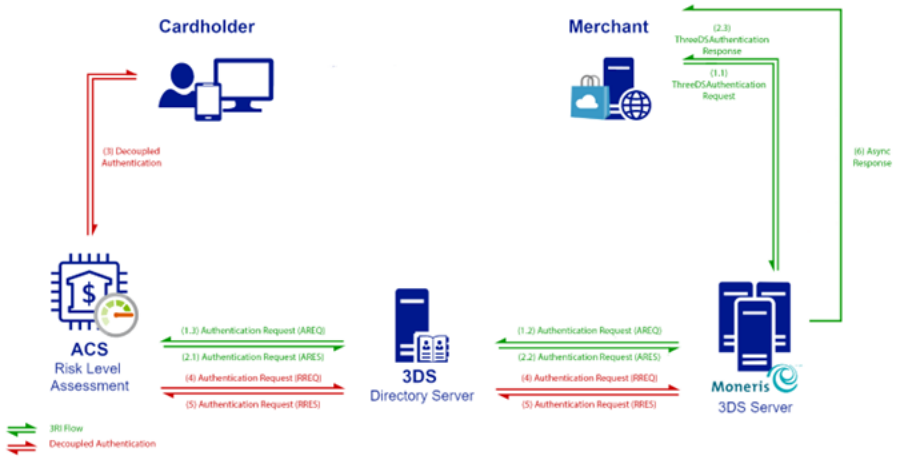
Steps 5 – 10 (Challenge Only)

In scenarios where a challenge is required, the merchant website sends an HTTP POST to the Challenge URL with the ChallengeData sent as a “CREQ” value (5). The ACS system will present a challenge to the cardholder, who will supply whatever credentials their issuer requires. The merchant website receives a

"CRES" value from the ACS via the HTTP POST response (6). Meanwhile, the ACS supplies the results to the 3DS Directory, which then forwards it to Moneris (7-8).

The merchant's website then sends a CAVV Lookup to Moneris via a cavv_lookup request and includes their "CRES" (9). Moneris responses with the cavv_lookup response with the necessary ECI and CAVV values. With the 3DS authentication complete, you can proceed to the financial transaction.

5.2.3 Transaction Flow for 3-D Secure - 3RI channel



In a 3DS Requestor Initiated flow, the cardholder is not directly triggering the transaction flow via a browser experience as above. It is possible they are initiating the transaction outside the 3DS protocol, such as mailing or phoning the merchant (Mail-Telephone Order, aka MOTO), or it is possible the merchant is processing a recurring or installment plan on behalf of the cardholder's subscription. It is also possible the merchant requires a non-payment authentication as part of tokenizing the card for later use.

3RI flows do not have direct cardholder interaction. The merchant sends their <threeDSAuthentication> request per steps 3-4 above but include additional fields to describe their 3RI usage scenario.

- If this is a Mail or Telephone (MOTO) payment authentication, the ACS may trigger a Decoupled Authentication between the issuer and cardholder (see Decoupled Authentication)
- If this is a follow-on payment from a previous 3DS authenticated transaction, you can include prior_request_auth_info to link to the previous authentication and improve the likelihood of a successful result

Your server can utilize the fields device_channel , ri_indicator and message_category to inform Moneris if your merchant server is attempting to use the 3DS Requestor Initiated process.

5.2.3.1 Decoupled Authentication

For scenarios where a 3RI authentication requires challenge, instead of utilizing the standard challenge request and response the ACS authenticates the cardholder outside of the 3-D Secure protocol such as a banking app or mobile phone text to the cardholder. The Moneris 3DS Server waits for the ACS to authenticate the cardholder; this authentication can take up to 7 days. As this process relies on a cardholder action outside the 3DS flow, it occurs asynchronously to transaction processing.

Your server can utilize the fields decoupled_request_indicator and decoupled_request_async_url to inform Moneris that you are opting in to accept a Decoupled Authentication attempt and where you want Moneris to POST the results asynchronously.

5.2.4 Mpi2Request Object and 3-D Secure Authentication

The authentication transactions for 3-D Secure 2.2 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 - Browser Channel
- 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.

5.2.5 3-D Secure 2.2 Request DTD

```
<!ELEMENT Mpi2Request (store_id, api_token, (card_lookup | threeds_authentication | cavv_
lookup))>

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

//Browser Channel only
<!ELEMENT threeds_authentication (message_category, device_channel, request_type, order_id,
(pan | data_key, expdate), amount, currency?, cardholder_name, threeds_completion_ind, bill_
address1, bill_province, bill_city, bill_postal_code, bill_country, ship_address1, ship_
province, ship_city, ship_postal_code, ship_country, notification_url, challenge_windowsize,
browser_useragent, browser_java_enabled, browser_screen_height, browser_screenwidth, browser_
language, email?, request_challenge?)>

//3RI, non-recurring
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_request_
indicator?, decoupled_request_max_time?, decoupled_request_async_url?, ri_indicator, prior_
authentication_info?, order_id, (pan | data_key, expdate), amount, currency?, cardholder_
```

```
name), bill_address1, bill_province, bill_city, bill_postal_code, bill_country, ship_
address1, ship_province, ship_city, ship_postal_code, ship_country,>

//3RI, recurring
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_request_
indicator?, decoupled_request_max_time?, decoupled_request_async_url?, recurring_frequency,
recurring_expiry, ri_indicator, prior_authentication_info, order_id, (pan | data_key,
expdate), amount, currency?, cardholder_name), bill_address1, bill_province, bill_city, bill_
postal_code, bill_country, ship_address1, ship_province, ship_city, ship_postal_code, ship_
country,>

<!ELEMENT prior_authentication_info (prior_request_auth_data, prior_request_ref, prior_
request_auth_method>)

<!ELEMENT cavv_lookup (cres)>

<!-- start 3DS 2.2 specific fields -->
<!ELEMENT threeds_version (#PCDATA)>
<!ELEMENT threeds_server_trans_id (#PCDATA)>
<!ELEMENT data_key (#PCDATA)>
<!ELEMENT notification_url (#PCDATA)>
<!ELEMENT cardholder_name (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT threeds_completion_ind (#PCDATA)>
<!ELEMENT request_type (#PCDATA)>
<!ELEMENT purchase_date (#PCDATA)>
<!ELEMENT challenge_windowsize (#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)>
<!ELEMENT message_category (#PCDATA)>
<!ELEMENT device_channel (#PCDATA)>
<!ELEMENT decoupled_request_indicator (#PCDATA)>
<!ELEMENT decoupled_request_max_time (#PCDATA)>
<!ELEMENT decoupled_request_async_url (#PCDATA)>
<!ELEMENT recurring_frequency (#PCDATA)>
<!ELEMENT recurring_expiry (#PCDATA)>
<!ELEMENT ri_indicator (#PCDATA)>
<!ELEMENT prior_request_auth_data (#PCDATA)>
<!ELEMENT prior_request_ref (#PCDATA)>
<!ELEMENT prior_request_auth_method (#PCDATA)>
<!ELEMENT prior_request_auth_timestamp (#PCDATA)>
```

5.2.6 3-D Secure 2.2 Response DTD

```
<!-- The following are only applicable to 3DS 2.2 transactions -->
<!ELEMENT MessageType (#PCDATA)>
<!ELEMENT ThreeDSResponseCode (#PCDATA)>
<!ELEMENT ThreeDSMessage (#PCDATA)>
<!ELEMENT ReceiptID (#PCDATA)>
```

```
<!ELEMENT ThreeDSMethodURL (#PCDATA)>
<!ELEMENT ThreeDSMethodData (#PCDATA)>
<!ELEMENT ChallengeURL (#PCDATA)>
<!ELEMENT ChallengeData (#PCDATA)>
<!ELEMENT ChallengeCompletionIndicator (#PCDATA)>
<!ELEMENT TransStatus (#PCDATA)>
<!ELEMENT ThreeDSServerTransId (#PCDATA)>
<!ELEMENT ECI (#PCDATA)>
<!ELEMENT Cavv (#PCDATA)>
<!ELEMENT ThreeDSVersion (#PCDATA)>
<!ELEMENT DSTransId (#PCDATA)>
<!ELEMENT TransStatusReason (#PCDATA)>
<!ELEMENT Cardholder Info (#PCDATA)>
<!ELEMENT AuthenticationType (#PCDATA)>
<!ELEMENT ThreeDSAcsTransID (#PCDATA)>
<!ELEMENT ThreeDSAuthTimeStamp (#PCDATA)>
```

5.3 Implementing Card Lookup Request

The CardLookup request verifies the applicability of 3DS 2.2 on the card and returns the 3DS Method URL used for device fingerprinting if the card supports this feature. 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, if supported.

- If you receive the threeDSMethodURL, you may send the threeDSMethodData to the threeDSMethodURL via a browser post in order to supplement the authentication request with device data pertaining to the cardholder's browser.

- If you do not receive the threeDSMethodURL, you may still proceed with 3DS Authentication.

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	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>	*Boolean* 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

Cavv Lookup Request 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.
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. Carries the token for network tokenization transactions. **NOTE:** Conditional. Either a credit card number or data key is required.
data key <data_key>	*String* 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 **NOTE:** Conditional. Either a credit card number or data key is required.

Variable Name	Type and Limits	Description
notification URL <notification_url>	*String* 256-character alpha-numeric	Notification URL for receiving the 3DS Method POST response from the issuer ACS.

5.4 Handling the 3DS Method for Device Fingerprinting

You can use the **threeDSMethodURL** & **threeDSMethodData** returned by a Card Lookup response to increase the probability of a frictionless 3DS flow for the cardholder. Transmitting the **threeDSMethodData** to the **threeDSMethodURL** via a browser HTTP POST allows the issuer use of a hidden iFrame on the merchant website to obtain details on the customer's device.

The results of the 3DS Method are returned to the merchant's **notificationURL** supplied in the preceding Card Lookup.

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

Device Fingerprinting request form (Merchant browser to ACS):

```
<form name="frm" method="POST" action="Rendering URL">
<input type="hidden" name="threeDSMethodData" value-
="eyJ0aHJlZURTU2VydmVyVHJh-
bnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OTFiLTJhYzA1YTU0MmM0YSIsInRocmVlRFNNZ-
XRob2ROb3RpZmljYXRpb25VUkwiOiJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIn0">
</form>
```

Decoded threeDSMethodData:

```
{"threeDSServerTransID":"3ac7caa7-aa42-2663-791b-2ac05a542c4a","-
threeDSMethodNotificationURL":"threeDSMethodNotificationURL"}
```

Device Fingerprinting response form (ACS to Merchant notificationURL):

```
<form name="frm" method="POST" action="threeDSMethodNotificationURL">
<input type="hidden" name="threeDSMethodData" value-
="eyJ0aHJlZURTU2VydmVyVHJh-
hbnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OTFiLTJhYzA1YTU0MmM0YSJ9">
</form>
```

Decoded threeDSMethodData:

```
{"threeDSServerTransID":"3ac7caa7-aa42-2663-791b-2ac05a542c4a"}
```

5.5 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 2.2 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 URLs:

https://mpg1t.moneris.io/mpi2/servlet/MpiServlet

Production URLs:

https://mpg1.moneris.io/mpi2/servlet/MpiServlet

Below we detail three different scenarios for utilizing Moneris MPI 3DS Authentication. Each scenario has conditions for which fields are required or optional for the endpoint.

5.5.1 MPI 3DS Authentication Request - Browser Channel

NOTE: Billing address request fields are recommended to be sent for this transaction, or else the authentication process may fail

XML transaction object

```
<threeds_authentication>
```

MPI 3DS Authentication Request transaction object definition

```
<!ELEMENT threeds_authentication (message_category, device_channel, request_
type, order_id, (pan | data_key, expdate), amount, currency?, cardholder_name,
threeds_completion_ind, bill_address1, bill_province, bill_city, bill_postal_
code, bill_country, ship_address1, ship_province, ship_city, ship_postal_code,
ship_country, notification_url, challenge_windowsize, browser_ip, browser_
useragent, browser_java_enabled, browser_screen_height, browser_screenwidth,
browser_language, email?, request_challenge?, work_phone, home_phone, mobile_
phone)>
```

WARNING: Do not send fields related to 3RI on browser-based authentications.

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	*String*	Unique identifier provided by Moneris

Variable Name	Type and Limits	Description
<store_id>	N/A	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>	*Boolean* 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

MPI 3DS Authentication Request transaction request fields – Required

Variable Name	Type and Limits	Description
message category <message_category>	*String* 2-character numeric	Whether the authentication request is for a payment or non-payment use: 01 = payment authentication (PA)

Variable Name	Type and Limits	Description
		02 = non-payment authentication (NPA)
device channel <device_channel>	*String* 2-character numeric	The interface used to initiate the authentication: 02 = Browser (BRW) 03 = 3DS Requestor Initiated (3RI)
request type <request_type>	*String* 2-character alphanumeric	Indicates the type of browser-based authentication request: 01 = cardholder initiated payment 02 = recurring transaction Conditional. Required if device_channel = 02
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.
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

Variable Name	Type and Limits	Description
		—field can be maximum 20 digits in support of future expansion of card number ranges. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
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
cardholder name <cardholder_name>	*String* 45-character alphanumeric **NOTE:** Accented characters are not allowable	Name of the cardholder
3DS completion indicator <three_ds_completion_ind>	*String* 1-character alphabetic	indicates whether 3ds method MpiCardLookup was successfully com-pleted Allowable values: Y = Successfully completed N = Did not successfully complete U = Unavailable Conditional. Required if card_lookup is

Variable Name	Type and Limits	Description
		used.
billing address <bill_address1>	*String* 50-character alphanumeric	Cardholder billing address
billing province <bill_province>	*String* 3-character alphanumeric	Cardholder province or state Defined in country subdivision ISO 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 alphanumeric	Cardholder billing country 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	Shipping destination province or state 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 or ZIP code
shipping country <ship_country>	*String* 3-character alphanumeric	Shipping destination country Defined as 3-digit country code in ISO 3166-1

Variable Name	Type and Limits	Description
notification URL <notification_url>	*String* 256-character alpha-numeric	Notification URL for receiving the 3DS Method POST response from the issuer ACS. Conditional. Required if device_channel = 02
challenge window size <challengeWindowsize>	*String* 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 Conditional. Required if device_channel = 02
browser IP Address <browser_ip_address>	*String* Allows '.' and ':' 45-character alphanumeric	IP address of the browser as returned by the HTTP headers to the 3DS Requestor. **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide. Lack of providing this field, might increase the risk of rejects.
browser user agent <browser_user_agent>	*String* 2048-character alpha-numeric	Browser User Agent Conditional. Required if device_channel = 02
browser java enabled <browser_java_enabled>	*String* 1-character alphabetic	Indicates whether Java is enabled in the browser Allowable values: T = True F = False

Variable Name	Type and Limits	Description
		Conditional. Required if device_channel = 02
browser screen height <browser_screen_height>	*String* 6-character numeric	Pixel height of cardholder screen Conditional. Required if device_channel = 02 **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide. Lack of providing this field, might increase the risk of rejects.
browser screen width <browser_screen_width>	*String* 6-character numeric	Pixel width of cardholder screen Conditional. Required if device_channel = 02 **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide. Lack of providing this field, might increase the risk of rejects.
browser language <browser_language>	*String* 8-character alphanumeric	As defined in IETF BCP47 Conditional. Required if device_channel = 02
email <email>	*String* 254-character alphanumeric	Cardholder email address **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide the cardholder's email address. Lack of providing the cardholder's address, might increase the risk of rejects.
cardholder work phone number <[[[Undefined variable Global-Request-Fields/Global-3DS-Req-Flds.Cardholder-Phone-Work]]]>	*Object* N/A	Cardholder work phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects.

Variable Name	Type and Limits	Description
		NOTE: This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.
cardholder home phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Home]]]>	*Object* N/A	Cardholder home phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects. **NOTE:** This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.
cardholder mobile phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Mobile]]]>	*Object* N/A	Cardholder mobile phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects. **NOTE:** This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.

MPI 3DS Cardholder Phone Number

Variable Name	Type and Limits	Description
country code <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-CountryCode]]]>	*String* 3-character numeric	Country Code of phone number provided by the Cardholder.
phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Subscriber]]]>	*String* 15-character numeric	The phone number provided by the Cardholder.

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 **NOTE:** This field should not be sent unless Multi Currency Pricing is enabled on your merchant account
request challenge <request_challenge>	*String* 2-character numeric	Indicates whether a browser-based challenge is requested for this transaction. Standard is “01” • 01 = No preference • 02 = No challenge requested

Variable Name	Type and Limits	Description
		• 03 = Challenge requested: 3DS Requestor Preference • 04 = Challenge requested: Mandate Conditional. Required if device_channel = 02

Sample MPI 3DS Authentication Request - Browser Channel

```
<Mpi2Request>
<store_id>store5</store_id>
<api_token>yesguy<api_token>
<threeds_authentication>
        <message_category>01</message_category>
        <device_channel>02</device_channel>
        <order_id>test authentication 2</order_id>
        <pan>4000###########0013</pan>
        <expdate>2105</expdate>
        <amount>1.00</amount>
        <threeds_completion_ind>Y</threeds_completion_ind>
        <request_type>01</request_type>
        <notification_url>https://merchant.example..com/notify.html</notification_url>
        <challenge_windowsize>03</challenge_windowsize>
        <cardholder_name>John Smith</cardholder_name>
        <bill_address1>1 Main St</bill_address1>
        <bill_province>ON</bill_province>
        <bill_city>Toronto</bill_city>
        <bill_postal_code>M8X 2X2</bill_postal_code>
        <bill_country>124</bill_country>
        <ship_address1>1 Main St</ship_address1>
        <ship_province>ON</ship_province>
        <ship_city>Toronto</ship_city>
        <ship_postal_code>M1M1M1</ship_postal_code>
        <ship_country>124</ship_country>
        <browser_ip>10.10.10.10</browser_ip>
        <browser_useragent>Mozilla/5.0 (Windows NT 10.0.....</browser_useragent>
        <browser_java_enabled>true</browser_java_enabled>
        <browser_screen_height>1000</browser_screen_height>
        <browser_screen_width>1920</browser_screen_width>
        <browser_language>en-GB</browser_language>
```

```
        <email>user@example.com</email>
        <work_phone>
                <cc>1</cc>
                <subscriber>1234567890</subscriber>
        </work_phone>
        <home_phone>
                <cc>1</cc>
                <subscriber>1234567890</subscriber>
        </home_phone>
        <mobile_phone>
                <cc>1</cc>
                <subscriber>1234567890</subscriber>
        </mobile_phone>
        <request_challenge>01</request_challenge>
</threeds_authentication>
</Mpi2Response>
```

Sample MPI 3DS Authentication Response - Frictionless (without Challenge)

```
<?xml version="1.0"?>
<Mpi2Response>
        <receipt>
                <MessageType>ARes</MessageType>
                <ResponseCode>001</ResponseCode>
                <Message>SUCCESS</Message>
                <ReceiptId>danlookup100666</ReceiptId>
                <ThreeDSMethodURL></ThreeDSMethodURL>
                <ThreeDSMethodData></ThreeDSMethodData>
                <ChallengeURL></ChallengeURL>
                <ChallengeData></ChallengeData>
                <TransStatus>Y</TransStatus>
                <ThreeDSServerTransId>b166120b-09d1-4b82-a343-a235e1ad6980</ThreeDSServerTransId>
                <DSTransId>aa5a81e5-4fe0-41e5-9e36-4b0187c4524e</DSTransId>
                <ECI>5</ECI>
                <Cavv>kAABApFSYyd4l2eQQFJjAAAAAAA=</Cavv>
                <TransStatusReason></TransStatusReason>
                <CardholderInfo></CardholderInfo>
                <ThreeDSVersion>2.2</ThreeDSVersion>
                <AuthenticationType></AuthenticationType>
                <ThreeDSAcsTransID>da49dc91-2f94-4c4a-bcaa-9700b9d7b205</ThreeDSAcsTransID>
                <ThreeDSAuthTimeStamp>201710282113</ThreeDSAuthTimeStamp>
    </receipt>
</Mpi2Response>
```

Sample MPI 3DS Authentication Response - with Challenge

```
<?xml version="1.0"?>
<Mpi2Response>
       <receipt>
              <MessageType>ARes</MessageType>
              <ResponseCode>001</ResponseCode>
              <Message>SUCCESS</Message>
              <ReceiptId>danlookup100666</ReceiptId>
              <ThreeDSMethodURL></ThreeDSMethodURL>
              <ThreeDSMethodData></ThreeDSMethodData>
              <ChallengeURL>https://acs-server.ps.msignia.com/api/v1/browser_challenges</ChallengeURL>
              <ChallengeData>eyJhY3NUcmFu....</ChallengeData>
              <TransStatus>C</TransStatus>
              <ThreeDSServerTransId>07834e85-f422-4565-ae02-a6a7dc1b3e84</ThreeDSServerTransId>
              <DSTransId>6330bf76-c7b2-4174-b291-20a6650d0b0f</DSTransId>
              <ECI></ECI>
              <Cavv></Cavv>
              <TransStatusReason></TransStatusReason>
              <CardholderInfo></CardholderInfo>
              <ThreeDSVersion>2.2</ThreeDSVersion>
              <AuthenticationType>01</AuthenticationType>
              <ThreeDSAcsTransID>da49dc91-2f94-4c4a-bcaa-9700b9d7b205</ThreeDSAcsTransID>
              <ThreeDSAuthTimeStamp></ThreeDSAuthTimeStamp>
    </receipt>
</Mpi2Response>
```

5.5.2 MPI 3DS Authentication Request - 3RI with recurring

> **NOTE:** Billing address request fields are recommended to be sent for this transaction, or else the authentication process may fail

XML transaction object

```
<threeds_authentication>
```

MPI 3DS Authentication Request transaction object definition

```
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_
request_indicator?, decoupled_request_max_time?, decoupled_request_async_url?,
```

```
recurring_frequency, recurring_expiry, ri_indicator, prior_authentication_
info, order_id, (pan | data_key, expdate), amount, currency?, cardholder_
name), bill_address1, bill_province, bill_city, bill_postal_code, bill_
country, ship_address1, ship_province, ship_city, ship_postal_code, ship_
country)>

<!ELEMENT prior_authentication_info (prior_request_auth_data, prior_request_
ref, prior_request_auth_method, prior_request_auth_timestamp)>
```

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>	*Boolean* 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

Variable Name	Type and Limits	Description
		investigation is required

MPI 3DS Authentication Request transaction request fields – Required

Variable Name	Type and Limits	Description
message category <message_category>	*String* 2-character numeric	Whether the authentication request is for a payment or non-payment use: 01 = payment authentication (PA) 02 = non-payment authentication (NPA)
device channel <device_channel>	*String* 2-character numeric	The interface used to initiate the authentication: 02 = Browser (BRW) 03 = 3DS Requestor Initiated (3RI)
recurring frequency <recurring_frequency>	*String* 4-character numeric	The minimum number of days between recurring transactions. Numeric values between 1 and 9999, leading zeroes accepted. Conditional.

Variable Name	Type and Limits	Description
		Required if request_type = 02 Conditional. Required if ri_ indicator = 01
recurring expiry <recurring_expiry>	*String* 8-character numeric	End date after which no further recurring trans-actions shall be performed. Format is YYYYMMDD. Conditional. Required if ri_ indicator = 01
ri indicator <ri_indicator> **NOTE:** Visa Secure only sup-port ri_Indicator = 01, 02, 06, 07, or 11 for Payment Transactions and ri Indicator = 03, 04, 05 and 10 for Non Payment Transactions	*String* 2-character numeric	The type of 3DS Requestor Ini-tiated (3RI) request: 01 = Recurring 02 = Installment 03 = Add Card 04 = Maintain Card Information 05 = Account veri-fication 06 = Split/Delayed Shipment 07 = Top-up 08 = Mail Order 09 = Telephone Order 10 = Whitelist

Variable Name	Type and Limits	Description
		11 = Other Payment Conditional. Required if device_channel = 03
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.
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

Variable Name	Type and Limits	Description
		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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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

Variable Name	Type and Limits	Description
		Minimum allowable value = $0.01, maximum allowable value = $9999999.99
cardholder name <cardholder_name>	*String* 45-character alphanumeric **NOTE:** Accented characters are not allowable	Name of the cardholder
billing address <bill_address1>	*String* 50-character alphanumeric	Cardholder billing address
billing province <bill_province>	*String* 3-character alphanumeric	Cardholder province or state Defined in country subdivision ISO 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 alphanumeric	Cardholder billing country 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	Shipping destination province or state

Variable Name	Type and Limits	Description
		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 or ZIP code
shipping country <ship_country>	*String* 3-character alphanumeric	Shipping destination country Defined as 3-digit country code in ISO 3166-1
email <email>	*String* 254-character alphanumeric	Cardholder email address **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide the cardholder's email address. Lack of providing the cardholder's address, might increase the risk of rejects.

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

Variable Name	Type and Limits	Description
		NOTE: This field should not be sent unless Multi Currency Pricing is enabled on your merchant account
decoupled request indicator <decoupled_ request_indicator>	*String* 1-character alphabetic	Whether the request utilizes Decoupled Authentication or not, if the ACS confirms its use. Y = Decoupled Authentication is supported and preferred if challenge is necessary N = Do not use Decoupled Authentication (Default) Defaults to N if unused.
decoupled request max time <decoupled_ request_max_time>	*String* 5-character numeric	The maximum minutes that Moneris waits for an ACS to provide results. Numeric values between 1 and 10080. The max is equivalent to 7 days. Conditional. Required if device_channel = 03 and decoupled_request_indicator = Y
decoupled request async URL <decoupled_ request_async_url>	*String* 256-character alphanumeric	Your URL where Moneris will POST the response back from ACS. Moneris reat-tempts 3 times to POST the response. Conditional. Only sent if decoupled-_ request-_indicator = Y
prior request auth info <prior_request_ auth_info>	*Object* N/A	Object containing details for a prior 3DS authentication for this series of trans-actions.This is a nested object within the authentication transaction, and required when storing or using the information about the prior authentication for that card. For information about fields in the Prior Authentication Info object, see MPI 3DS Prior Authentication Info Object and Variables.

MPI 3DS Prior Authentication Info

Variable Name	Type and Limits	Description
prior request auth data <prior_request_auth_data>	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.
prior request ref <prior_request_auth_ref>	*String* 36-character alphanumeric	Refers to the 3DS ACS Transaction ID in the response of the previous 3DS authentication.
prior request auth method <prior_request_auth_method>	*String* 2-character numeric	Mechanism used by the cardholder to authenticate in the previous 3DS authentication: 01 = Frictionless authentication 02 = Challenge authentication 03 = AVS verified 04 = Other issuer methods
prior request auth timestamp <prior_request_auth_timestamp>	*String* 12-character numeric	Date and time in UTC of the prior cardholder authentication. Found in the previous 3DS authentication response as 3DS Auth TimeStamp. Format is YYYYMMDDHHMM.

Sample MPI 3DS Authentication Request - 3RI with recurring

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpi2Request>
	<store_id>store5</store_id>
	<api_token>yesguy</api_token>
	<threeds_authentication>
		<message_category>01</message_category>
		<device_channel>03</device_channel>
		<decoupled_request_max_time>10080</decoupled_request_max_time>
		<decoupled_request_indicator>Y</decoupled_request_indicator>
		<decoupled_request_async_url>my.server.com</decoupled_request_async_url>
		<ri_indicator>01</ri_indicator>
		<prior_authenitcation_info>
```

```
                        <prior_request_ref>d7c1ee99-9478-44a6-b1f2-391e29c6b340</prior_request_ref>
                        <prior_request_auth_data>abcdabdc…</prior_request_auth_data>
                        <prior_request_auth_method>01</prior_request_auth_method>
                        <prior_request_auth_timestamp>201710282113</prior_request_auth_timestamp>
                </prior_authenitcation_info>
                <order_id>danlookup100666</order_id>
                <cardholder_name>John Smith</cardholder_name>
                <pan>4606633870436092</pan>
                <expdate>2105</expdate>
                <amount>1.00</amount>
                <threeds_completion_ind>Y</threeds_completion_ind>
                <request_type>02</request_type>
                <recurring_frequency>31</recurring_frequency>
                <recurring_expiry>20231231</recurring_expiry>
        </threeds_authentication>
</Mpi2Request>
```

Sample MPI 3DS Authentication Response

```
<?xml version="1.0"?>
<Mpi2Response>
        <receipt>
                <MessageType>ARes</MessageType>
                <ResponseCode>001</ResponseCode>
                <Message>SUCCESS</Message>
                <ReceiptId>danlookup10066633333333333</ReceiptId>
                <ThreeDSMethodURL></ThreeDSMethodURL>
                <ThreeDSMethodData></ThreeDSMethodData>
                <ChallengeURL></ChallengeURL>
                <ChallengeData></ChallengeData>
                <TransStatus>Y</TransStatus>
                <ThreeDSServerTransId>a3aa9295-7ce0-4856-a969-4951bb4b9310
                </ThreeDSServerTransId>
                <DSTransId>88632ac9-5873-4cf2-9637-b9cd8006e359</DSTransId>
                <ECI>5</ECI>
                <Cavv>kAABApFSYyd4l2eQQFJjAAAAAAA=</Cavv>
                <TransStatusReason></TransStatusReason>
                <CardholderInfo></CardholderInfo>
                <ThreeDSVersion></ThreeDSVersion>
                <AuthenticationType>04</AuthenticationType>
                <ThreeDSAcsTransID>da49dc91-2f94-4c4a-bcaa-9700b9d7b205</ThreeDSAcsTransID>
                <ThreeDSAuthTimeStamp>201710282113</ThreeDSAuthTimeStamp>
```

```
      </receipt>
</Mpi2Response>
```

5.5.3 MPI 3DS Authentication Request - 3RI, non-recurring

NOTE: Billing address request fields are recommended to be sent for this transaction, or else the authentication process may fail

XML transaction object

```
<threeds_authentication>
```

MPI 3DS Authentication Request transaction object definition

```
<!ELEMENT threeds_authentication (message_category, device_channel, decoupled_
request_indicator?, decoupled_request_max_time?, decoupled_request_async_url?,
ri_indicator, prior_authentication_info?, order_id, (pan | data_key, expdate),
amount, currency?, cardholder_name), bill_address1, bill_province, bill_city,
bill_postal_code, bill_country, ship_address1, ship_province, ship_city, ship_
postal_code, ship_country>

<!ELEMENT prior_authentication_info (prior_request_auth_data, prior_request_
ref, prior_request_auth_method>)
```

WARNING: Do not send fields related to 3RI on browser-based authentications.

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:

Variable Name	Type and Limits	Description
		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>	*Boolean* 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

MPI 3DS Authentication Request transaction request fields – Required

Variable Name	Type and Limits	Description
message category <message_category>	*String* 2-character numeric	Whether the authentication request is for a payment or non-payment use: 01 = payment authentication (PA) 02 = non-payment authentication (NPA)
device channel <device_channel>	*String* 2-character numeric	The interface used to initiate the authentication: 02 = Browser (BRW) 03 = 3DS Requestor Initiated (3RI)

Variable Name	Type and Limits	Description
ri indicator <ri_indicator> **NOTE:** Visa Secure only support ri_Indicator = 01, 02, 06, 07, or 11 for Payment Transactions and ri Indicator = 03, 04, 05 and 10 for Non Payment Transactions	*String* 2-character numeric	The type of 3DS Requestor Initiated (3RI) request: 01 = Recurring 02 = Installment 03 = Add Card 04 = Maintain Card Information 05 = Account verification 06 = Split/Delayed Shipment 07 = Top-up 08 = Mail Order 09 = Telephone Order 10 = Whitelist 11 = Other Payment
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.
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

Variable Name	Type and Limits	Description
		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. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
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
cardholder name <cardholder_name>	*String* 45-character alphanumeric **NOTE:** Accented characters are not allowable	Name of the cardholder
billing address <bill_address1>	*String* 50-character alphanumeric	Cardholder billing address
billing province <bill_province>	*String* 3-character alphanumeric	Cardholder province or state Defined in country subdivision ISO 3166-2

Variable Name	Type and Limits	Description
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 alphanumeric	Cardholder billing country 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	Shipping destination province or state 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 or ZIP code
shipping country <ship_country>	*String* 3-character alphanumeric	Shipping destination country Defined as 3-digit country code in ISO 3166-1
email <email>	*String* 254-character alpha-numeric	Cardholder email address **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide the cardholder's email address. Lack of providing the cardholder's address, might increase the risk of rejects.

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 **NOTE:** This field should not be sent unless Multi Currency Pricing is enabled on your merchant account
decoupled request indicator <decoupled_ request_indicator>	*String* 1-character alphabetic	Whether the request utilizes Decoupled Authentication or not, if the ACS confirms its use. Y = Decoupled Authentication is supported and preferred if challenge is necessary N = Do not use Decoupled Authentication (Default) Defaults to N if unused.
decoupled request max time <decoupled_ request_max_time>	*String* 5-character numeric	The maximum minutes that Moneris waits for an ACS to provide results. Numeric values between 1 and 10080. The max is equivalent to 7 days. Conditional. Required if device_channel = 03 and decoupled_request_indicator = Y
decoupled request async URL <decoupled_ request_async_url>	*String* 256-character alphanumeric	Your URL where Moneris will POST the response back from ACS. Moneris reat-tempts 3 times to POST the response. Conditional. Only sent if decoupled-_ request-_indicator = Y
prior request auth info <prior_request_ auth_info>	*Object* N/A	Object containing details for a prior 3DS authentication for this series of trans-actions.This is a nested object within the authentication transaction, and required when storing or using the information about the prior authentication for that card. For information about fields in the

Variable Name	Type and Limits	Description
		Prior Authentication Info object, see MPI 3DS Prior Authentication Info Object and Variables.

MPI 3DS Prior Authentication Info

Variable Name	Type and Limits	Description
prior request auth data <prior_request_auth_data>	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.
prior request ref <prior_request_auth_ref>	*String* 36-character alphanumeric	Refers to the 3DS ACS Transaction ID in the response of the previous 3DS authentication.
prior request auth method <prior_request_auth_method>	*String* 2-character numeric	Mechanism used by the cardholder to authenticate in the previous 3DS authentication: 01 = Frictionless authentication 02 = Challenge authentication 03 = AVS verified 04 = Other issuer methods
prior request auth timestamp <prior_request_auth_timestamp>	*String* 12-character numeric	Date and time in UTC of the prior cardholder authentication. Found in the previous 3DS authentication response as 3DS Auth TimeStamp. Format is YYYYMMDDHHMM.

Sample MPI 3DS Authentication Request - 3RI without recurring

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpi2Request>
<store_id>store5</store_id>
<api_token>yesguy</api_token>
<threeds_authentication>
       <message_category>01</message_category>
```

```
        <device_channel>03</device_channel>
        <decoupled_request_max_time>10080</decoupled_request_max_time>
        <decoupled_request_indicator>Y</decoupled_request_indicator>
        <decoupled_request_async_url>my.server.com</decoupled_request_async_url>
        <ri_indicator>08</ri_indicator>
        <prior_authenitcation_info>
                <prior_request_ref>d7c1ee99-9478-44a6-b1f2-391e29c6b340</prior_request_ref>
                <prior_request_auth_data>abcdabdc…</prior_request_auth_data>
                <prior_request_auth_method>01</prior_request_auth_method>
                <prior_request_auth_timestamp>201710282113</prior_request_auth_timestamp>
        </prior_authenitcation_info>
        <order_id>danlookup100666</order_id>
        <cardholder_name>John Smith</cardholder_name>
        <pan>4606633870436092</pan>
        <expdate>2105</expdate>
        <amount>1.00</amount>
        <threeds_completion_ind>Y</threeds_completion_ind>
        <request_type>01</request_type>
</threeds_authentication>
</Mpi2Request>
```

Sample MPI 3DS Authentication Response - 3RI, decoupled challenge

```
<?xml version="1.0"?>
<Mpi2Response>
    <receipt>
                <MessageType>ARes</MessageType>
                <ResponseCode>001</ResponseCode>
                <Message>SUCCESS</Message>
                <ReceiptId>danlookup10066633333333333</ReceiptId>
                <ThreeDSMethodURL></ThreeDSMethodURL>
                <ThreeDSMethodData></ThreeDSMethodData>
                <ChallengeURL></ChallengeURL>
                <ChallengeData></ChallengeData>
                <TransStatus>D</TransStatus>
                <ThreeDSServerTransId>a3aa9295-7ce0-4856-a969-4951bb4b9310</ThreeDSServerTransId>
                <DSTransId>88632ac9-5873-4cf2-9637-b9cd8006e359</DSTransId>
                <ECI></ECI>
                <Cavv></Cavv>
                <TransStatusReason></TransStatusReason>
                <CardholderInfo></CardholderInfo>
                <ThreeDSVersion></ThreeDSVersion>
                <AuthenticationType></AuthenticationType>
```

```
            <ThreeDSAcsTransID>da49dc91-2f94-4c4a-bcaa-9700b9d7b205</ThreeDSAcsTransID>
            <ThreeDSAuthTimeStamp></ThreeDSAuthTimeStamp>
    </receipt>
</Mpi2Response>
```

Sample MPI 3DS Authentication Response - 3RI, decoupled async response (second)

```
<?xml version="1.0"?>
<Mpi2Response>
       <receipt>
            <MessageType></MessageType>
            <ResponseCode>001</ResponseCode>
            <ReceiptId>O641172836</ReceiptId>
            <ThreeDSMethodURL></ThreeDSMethodURL>
            <ThreeDSMethodData></ThreeDSMethodData>
            <ChallengeURL></ChallengeURL>
            <ChallengeData></ChallengeData>
            <ChallengeCompletionIndicator></ChallengeCompletionIndicator>
            <TransStatus>Y</TransStatus>
            <ECI>5</ECI>
            <ThreeDSServerTransId>8ed21a5c-99c9-47ae-9e2e-5ab576de8768</ThreeDSServerTransId>
            <Cavv>AAICB5dnRwAAAEhEEkCQdAAAAAA=</Cavv>
            <Message>SUCCESS</Message>
            <TransStatusReason></TransStatusReason>
            <CardholderInfo></CardholderInfo>
            <ThreeDSVersion></ThreeDSVersion>
            <AuthenticationType></AuthenticationType>
            <ThreeDSAcsTransID>da49dc91-2f94-4c4a-bcaa-9700b9d7b205</ThreeDSAcsTransID>
            <ThreeDSAuthTimeStamp></ThreeDSAuthTimeStamp>
       </receipt>
</Mpi2Response>
```

5.6 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>
```

5.6.1 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	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>	*Boolean* 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

Cavv Lookup Request transaction request fields – Required

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

5.7 Handling the Decoupled Authentication Flow

If you get a TransStatus = “D” in your threeDSAuthentication Response, then your server must be prepared to accept a second asynchronous response from Moneris.

The cardholder will be engaged by their issuer for cardholder authentication outside the 3DS protocol. This may involve alternate authentication applications or SMS prompts to the cardholder to confirm.

The cardholder is given up to 7 days to complete this decoupled challenge. Once completed, the issuer will communicate to Moneris and our MPI system sends a second 3DS Authentication Response to the address you define in <decoupled_request_async_url>

Sample Authentication Decoupled Authentication Flow

```
xml version="1.0" encoding="UTF-8"?>
<Mpi2Request>
    <store_id>store5</store_id>
    <api_token>yesguy</api_token>
    <threeds_authentication>
```

```
        <message_category>01</message_category>
        <device_channel>03</device_channel>
        <decoupled_request_max_time>10080</decoupled_request_max_time>
        <decoupled_request_indicator>Y</decoupled_request_indicator>
        <decoupled_request_async_url></decoupled_request_async_url>
        <ri_indicator>08</ri_indicator>
        <prior_authenitcation_info>
            <prior_request_ref></prior_request_ref>
            <prior_request_auth_data></prior_request_auth_data>
            <prior_request_auth_method>01</prior_request_auth_method>
            <prior_request_auth_timestamp></prior_request_auth_timestamp>
        </prior_authenitcation_info>
        <order_id>danlookup100666</order_id>
        <cardholder_name>John Smith</cardholder_name>
        <pan>378364151363839</pan>
        <expdate>2105</expdate>
        <amount>1.00</amount>
        <threeds_completion_ind>Y</threeds_completion_ind>
        <request_type>01</request_type>
    </threeds_authentication>
</Mpi2Request>
```

5.8 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.

5.8.1 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.

In addition to 3-D Secure transactions, this transaction can also be used to process Apple Pay and Google Pay™ transactions.

For mobile wallets, this transaction is applicable only if choosing to integrate directly to Apple Wallet or Google Wallet (if not using the Moneris Apple Pay or Google Pay™ SDKs). Refer to Apple or Google developer portals for details on integrating directly to their wallets to retrieve the payload data.

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?, wallet_indicator?, cust_info?, avs_info?, cvd_info?,
recur?, cof_info?, pbb_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>	*Boolean* 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

Purchase with 3-D Secure transaction request fields – Required

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*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
credit card number <pan>	*String* 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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
Cardholder Authentication Verification Value (CAVV) <cavv>	*String* 50-character alphanumeric	Value provided by the Moneris MPI or by a third-party MPI Sent in all 3-D Secure transactions,

Variable Name	Type and Limits	Description
		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

3-D Secure 2.2 -specific fields – Required

Variable Name	Type and Limits	Description
3DS version <threeds_version> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID <threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction

Purchase with 3-D Secure 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

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
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass

Variable Name	Type and Limits	Description
		NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
foreign indicator <foreign_indicator>	*Boolean* true or false	Used to identify domestic transactions processed by a marketplace merchant that is in a different country.
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.

3-D Secure 2.2 -specific fields – Optional

Variable Name	Type and Limits	Description
DS transaction ID ds_trans_id	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.

Variable Name	Type and Limits	Description
NOTE: Only used in financial transactions using 3rd Party 3-D Secure services.		

5.8.2 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.

In addition to 3-D Secure transactions, this transaction can also be used to process Apple Pay and Google Pay™ transactions.

For mobile wallets, this transaction is applicable only if choosing to integrate directly to Apple Wallet or Google Wallet (if not using the Moneris Apple Pay or Google Pay™ SDKs). Refer to Apple or Google developer portals for details on integrating directly to their wallets to retrieve the payload data.

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?, wallet_indicator?, cust_info?, avs_info?, cvd_info?, cof_
info?, pbb_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>	*Boolean* 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

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

Variable Name	Type and Limits	Description
order ID <order_id>	*String* 50-character alpha-numerica-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>	*String* 10-character decimal Up to 7 digits (dollars) +	Transaction dollar amount This must contain at least 3 digits, two of which are penny values

Variable Name	Type and Limits	Description
	decimal point (.) + 2 digits (cents) after the decimal point **EXAMPLE:** 1234567.89	Minimum allowable value = $0.01, maximum allowable value = $9999999.99
credit card number <pan>	*String* 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. Carries the token for network token-ization transactions.
expiry date <expdate>	*String* 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.
Cardholder Authentication Verification Value (CAVV) <cavv>	*String* 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

3-D Secure 2.2 -specific fields – Required

Variable Name	Type and Limits	Description
3DS version <threeds_version>	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0

Variable Name	Type and Limits	Description
NOTE: Mandatory for financial transactions using 3rd Party 3-D Secure services.		2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID <threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction

Pre-Authorization with 3-D Secure 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 **NOTE:** The 22-character maximum limit must take the "/" into account as one of

Variable Name	Type and Limits	Description
		the characters
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
foreign indicator <foreign_indicator>	*Boolean* true or false	Used to identify domestic transactions processed by a marketplace merchant that is in a different country.
Customer Information	*Object*	Contains fields that describe mis-

Variable Name	Type and Limits	Description
<cust_info>	N/A	cellaneous customer information, billing and shipping information, and item information
is incremental is_incremental	*Boolean* true/false	Indicates if this preauthorization is using an estimated amount. Estimations allow for incrementing the amount held via subsequent incrementalAuth requests. Defaults to false. **NOTE:** Please note that if this field is true, the preauthorization is only eligible for a single Preauthorization Completion. Any completion sent for partial completion is treated as a full completion (ship_indicator= P is treated as = F when is_incremental= true on the original preauth)
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.

3-D Secure 2.2 -specific fields – Optional

Variable Name	Type and Limits	Description
DS transaction ID <ds_trans_id>	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.

Variable Name	Type and Limits	Description
NOTE: Only used in financial transactions using 3rd Party 3-D Secure services.		

5.9 Testing Your 3-D Secure 2.2 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

5.10 Moving to Production With 3-D Secure 2.2

Once you have finished testing your 3D Secure 2.2 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

5.11 3-D Secure 2.2 TransStatus Codes

Value	Description	Comments
Y	Authenticated	Cardholder has been fully authenticated
D	Challenge Required (Decoupled)	Cardholder requires a challenge using Decoupled Authentication
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
R	Not Authenticated	Not authenticated because the Issuer is rejecting authentication and requesting that authorisation not be attempted

5.12 3-D Secure 2.2 Commons TransStatusReason Decline Codes

The following codes are returned by the 3-D Secure service in order to provide additional information about the 3-D Secure transaction status.

TransStatusReason Code	Description
01	Card authentication failed
02	Unknown Device
03	Unsupported Device
04	Exceeds authentication frequency limit
05	Expired card
06	Invalid card number

TransStatusReason Code	Description
07	Invalid transaction
08	No Card record
09	Security failure
10	Stolen card
11	Suspected fraud
12	Transaction not permitted to cardholder
13	Cardholder not enrolled in service
14	Transaction timed out at the ACS
15	Low confidence
16	Medium confidence
17	High confidence
18	Very High confidence
19	Exceeds ACS maximum challenges
20	Non-Payment transaction not supported
21	3RI transaction not supported
22	ACS technical issue
23	Decoupled Authentication required by ACS but not requested by 3DS Requestor
24	3DS Requestor Decoupled Max Expiry Time exceeded
25	Decoupled Authentication was provided insufficient time to authenticate cardholder. ACS will not make attempt
26	Authentication attempted but not performed by

TransStatusReason Code	Description
	the cardholder

NOTE: For a list of all TransStatus Decline Codes, please see Reference section of 3D Secure 2.2 at https://developer.moneris.com.

5.13 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.

5.13.1 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 Secureprocess correctly, the liability for this

Result Code	Message	Significance to Merchants
		transaction should remain with the Issuer for chargeback reason codes covered by Visa Secure.
2	CAVV passed validation (authentication)	Fully authenticated transaction. There is a 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.

5.13.2 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.

Result Code	Message	Significance to Merchants
		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 chargebacks.

5.13.3 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

Result Code	Description
U	AEVV Unchecked

6 Installments by Visa

- 6.1 About Installments by Visa
- 6.2 Installments by Visa Transaction Types
- 6.3 Sending Transactions with Installments by Visa
- 6.4 Installment Plan Lookup
- 6.5 Vault Installment Plan Lookup
- 6.6 Installment Info Object

6.1 About Installments by Visa

Installments by Visa enables issuers the ability to offer cardholders installment payment plans at the time of purchase. When a cardholder accepts an installment plan option, the merchant receives the payment in full, and the cardholder pays the issuer according to the plan.

For a full list of definitions of the request and response fields see B.1 Definition of Response Fields – Installments by Visa

6.2 Installments by Visa Transaction Types

Financial transactions that support Installments by Visa include the following:

- Purchase
- Pre-Authorization
- Pre-Authorization Completion
- Purchase Correction
- Refund

- Purchase with Vault – res_purchase_cc

- Pre-Authorization with Vault – res_preauth_cc

NOTE: Independent Refund transactions do not support Installments by Visa

WARNING: Do not send the Installment Info object on any transaction that is not intended to offer Installments by Visa functionality; doing so may cause the transaction to fail.

6.3 Sending Transactions with Installments by Visa

Sending transactions with Installments by Visa functionality involves the following steps:

1. Send the Installment Plan Lookup or Vault Installment Plan Lookup (for Vault transactions) transaction request to obtain the **installment plan ID**, **installment plan reference** and **terms and conditions version** data in the response

2. Present the offered installment plan(s) to the cardholder and obtain their agreement to a particular plan.

3. Using the data obtained in the response above, send the Installment Info object in the Purchase or Pre-Authorization; for Vault transactions, use Purchase with Vault or Pre-Authorization with Vault

When completing the transaction with a Pre-Authorization Completion, or when doing a Purchase Correction or Refund, as in the rest of the Unified API, the previous transactions are referenced using the **order ID** and **transaction number**, or for Vault transactions, using the **data key**.

NOTE: Independent Refund transactions do not support Installments by Visa

6.4 Installment Plan Lookup

Used to obtain information required to do financial transactions with Installments by Visa.

Installment Plan Lookup transaction object definition

installmentLookup

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	*String* N/A	store_id
API token	*String* N/A	api_token

Installment Plan Lookup transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String* 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	order_id
amount	*String* 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point **EXAMPLE:** 1234567.89	amount
credit card number	*String* max 20-character alpha-numeric	pan
expiry date	*String* 4-character alphanumeric YYMM	expdate

6.5 Vault Installment Plan Lookup

Used to obtain information required to do financial transactions with installments when using a token stored in the Moneris Vault.

Vault Installment Plan Lookup transaction object definition

resInstallmentLookup

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	*String* N/A	store_id
API token	*String* N/A	api_token

Vault Installment Plan Lookup transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String* 50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	order_id
amount	*String* 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point **EXAMPLE:** 1234567.89	amount
data key	*String* 25-character alphanumeric	data_key
expiry date	*String* 4-character alphanumeric	expdate

Variable Name	Type and Limits	Description
NOTE: Only send this field if using a temporary token; if not, omit this field	YYMM	

6.6 Installment Info Object

When sending Purchase or Pre-Authorization transactions with Installments by Visa, the Installment Info object is included in the request. The Installment Info object uses information received in the response to the Installment Plan Lookup transaction.

For a full list of definitions of the request and response fields see B.1 Definition of Response Fields – Installments by Visa

Installment Info object request fields

Variable Name	Type and Limits	Description
installment plan ID	*String* 36-character alphanumeric fixed length	plan_id
installment plan reference	*String* 10-character alphanumeric fixed length	plan_id_ref
terms and conditions version	*String* 10-character alphanumeric variable length (1-10 characters)	tac_version

WARNING: Do not send the Installment Info object on any transaction that is not intended to offer Installments by Visa functionality; doing so may cause the transaction to fail.

7 Multi-Currency Pricing (MCP)

- 7.1 About Multi-Currency Pricing (MCP)
- 7.2 Methods of Processing MCP Transactions
- 7.3 Multi-Currency Pricing (MCP) Request DTD
- 7.4 Multi-Currency Pricing (MCP) Response DTD
- 7.5 MCP Purchase
- 7.6 MCP Purchase with 3-D Secure
- 7.7 MCP Purchase with 3-D Secure and Vault
- 7.8 MCP Pre-Authorization
- 7.9 MCP Pre-Authorization with 3-D Secure
- 7.10 MCP Pre-Authorization with 3-D Secure and Vault
- 7.11 MCP Pre-Authorization Completion
- 7.12 MCP Purchase Correction
- 7.13 MCP Refund
- 7.14 MCP Independent Refund
- 7.15 MCP Purchase With Vault
- 7.16 MCP Pre-Authorization With Vault
- 7.17 MCP Independent Refund with Vault
- 7.18 MCP Get Rate
- 7.19 MCP Currency Codes
- 7.20 MCP Error Codes

7.1 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

7.2 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

7.3 Multi-Currency Pricing (MCP) Request DTD

```
<!--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)>
<!--NOTE: threeds_version and threeds_server_trans_id are mandatory for 3DS Version 2.0+ -->

<!ELEMENT mcp_cavv_preauth (order_id , cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_server_trans_id, cust_info?,
avs_info?, cvd_info?, cof_info?, ds_trans_id?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?)>
```

```
<!ELEMENT mcp_cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv, crypt_type?,
dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_server_trans_id, cust_info?,
avs_info?, cvd_info?, recur?, cof_info?, ds_trans_id?, mcp_version, cardholder_amount,
cardholder_currency_code, mcp_rate_token?)>

<!ELEMENT mcp_cavv_res_preauth_cc (data_key, order_id, cust_id, crypt_type, dynamic_
descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_
token?, threeds_version, threeds_server_trans_id, ds_trans_id?)>

<!ELEMENT mcp_cavv_res_purchase_cc (data_key, order_id, cust_id, crypt_type, dynamic_
descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_currency_code, mcp_rate_
token?, threeds_version, threeds_server_trans_id, ds_trans_id?)>
```

7.4 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)>
```

7.5 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	*String*	Unique identifier provided by Moneris

Variable Name	Type and Limits	Description
<store_id>	N/A	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>	*Boolean* 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 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

Variable Name	Type and Limits	Description
		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 **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>	*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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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)

Variable Name	Type and Limits	Description
		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 = V, 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>	*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 Purchase transaction request fields – Optional

Variable Name	Type and Limits	Description
dynamic descriptor	*String*	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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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 **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
wallet indicator	*String*	
<wallet_indicator>	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, or via network tokenization from the card brand. Required for Apple Pay, 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

Variable Name	Type and Limits	Description
		GPP – Google Pay™ In-App GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
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

7.6 MCP Purchase with 3-D Secure

XML transaction object

<mcpCavvPurchase>

MCP Purchase with 3-D Securetransaction object definition

```
<!ELEMENT mcp_cavv_purchase (order_id, cust_id?, amount, pan, expdate, cavv,
crypt_type?, dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_
server_trans_id, cust_info?, avs_info?, cvd_info?, recur?, cof_info?, ds_
trans_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	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>	*Boolean* 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 Purchase with 3-D Secure 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.
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
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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
Cardholder Authentication Verification Value (CAVV) <cavv>	*String* 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

Variable Name	Type and Limits	Description
		For Purchase and Pre-Authorization transactions with 3-D Secure for Apple Pay and Google Pay, the CAVV field contains the decrypted cryptogram
3DS version <threeds_version> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID <threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction
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**

Variable Name	Type and Limits	Description
		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 = V, 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>	*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 Purchase with 3-D Secure 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	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name

Variable Name	Type and Limits	Description
	and separator **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported.

Variable Name	Type and Limits	Description
		NOTE: Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
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
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
Credential on File Info <cof_info> **NOTE:** This is a nested object within the transaction, and required when storing or using the customer's stored credentials. For information about fields in the Credential on FIle Info object, see Credential on File Info Object and Variables.	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
Recurring Billing <recur>	*Object* N/A	Contains fields related to Recurring Billing
DS transaction ID	*String*	Refers to the DSTransID in the response of the previous 3DS authen-

Variable Name	Type and Limits	Description
<ds_trans_id> **NOTE:** Only used in financial transactions using 3rd Party 3-D Secure services.	36-character alphanumeric	tication.

7.7 MCP Purchase with 3-D Secure and Vault

XML transaction object

<mcpResCavvPurchaseCC>

MCP Purchase with 3-D Secure and Vault transaction object definition

```
<!ELEMENT mcp_cavv_res_purchase_cc (data_key, order_id, cust_id, crypt_type,
dynamic_descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?, threeds_version, threeds_server_trans_id, ds_
trans_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	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>	*Boolean* 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 Purchase with 3-D Secure and Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
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	Merchant-defined field that can be used as an identifier

Variable Name	Type and Limits	Description
	NOTE: Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Searchable from the Moneris Merchant Resource Center
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. Carries the token for network tokenization transactions.
Cardholder Authentication Verification Value (CAVV) <data_key>	*String* 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
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**

Variable Name	Type and Limits	Description
		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 = V, 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>	*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
3DS version <threeds_version> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID	*String*	Data is obtained from a Cavv Lookup

Variable Name	Type and Limits	Description
<threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	36-character numeric	Request or MPI 3DS Authentication Request transaction

MCP Purchase with 3-D Secure and 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 **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>	*String* 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>	*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

Variable Name	Type and Limits	Description
		subsequent MCP financial transaction requests in order to redeem that rate
DS transaction ID <ds_trans_id> **NOTE:** Only used in financial transactions using 3rd Party 3-D Secure services.	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.

7.8 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	Unique alphanumeric string assigned by Moneris upon merchant account activation

Variable Name	Type and Limits	Description
		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>	*Boolean* 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 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.

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
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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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 allow-

Variable Name	Type and Limits	Description
		able 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 = V, 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>	*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 Pre-Authorization 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	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

Variable Name	Type and Limits	Description
	NOTE: Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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 **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
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass

Variable Name	Type and Limits	Description
		NOTE: Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
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

7.9 MCP Pre-Authorization with 3-D Secure

XML transaction object

<mcpCavvPreauth>

MCP Pre-Authorization with 3-D Secure transaction object definition

```
<!ELEMENT mcp_cavv_preauth (order_id , cust_id?, amount, pan, expdate, cavv,
crypt_type?, dynamic_descriptor?, wallet_indicator?, threeds_version, threeds_
server_trans_id, cust_info?, avs_info?, cvd_info?, cof_info?, ds_trans_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	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>	*Boolean* 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 with 3-D Secure transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
expiry date <expdate>	*String* 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.
Cardholder Authentication Verification Value (CAVV) <cavv>	*String* 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
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

Variable Name	Type and Limits	Description
		2 – Mail Order / Telephone Order—Recurring 3 – Mail Order / Telephone Order—Instalment 4 – Mail Order / Telephone Order—Unknown classification 5 – Authenticated e-commerce transaction (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 = V, 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
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
3DS version <threeds_version>	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0

Variable Name	Type and Limits	Description
NOTE: Mandatory for financial transactions using 3rd Party 3-D Secure services.		2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID <threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction

MCP Pre-Authorization with 3-D Secure 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
wallet indicator <wallet_indicator>	*String* 3-character alphanumeric	Indicates when a card number has been collected via a digital wallet, such as in Apple Pay, Google Pay™,

Variable Name	Type and Limits	Description
		Visa Checkout and Mastercard MasterPass, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
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
Customer Information <cust_info>	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information

Variable Name	Type and Limits	Description
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
Credential on File Info <cof_info> **NOTE:** This is a nested object within the transaction, and required when storing or using the customer's stored credentials. For information about fields in the Credential on FIle Info object, see Credential on File Info Object and Variables.	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.
DS transaction ID <ds_trans_id> **NOTE:** Only used in financial transactions using 3rd Party 3-D Secure services.	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.

7.10 MCP Pre-Authorization with 3-D Secure and Vault

XML transaction object

<mcpResCavvPreauthCC>

MCP Pre-Authorization with 3-D Secure and Vault transaction object definition

```
<!ELEMENT mcp_cavv_res_preauth_cc (data_key, order_id, cust_id, crypt_type,
dynamic_descriptor?, expdate?, mcp_version, cardholder_amount, cardholder_
currency_code, mcp_rate_token?, threeds_version, threeds_server_trans_id, ds_
trans_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	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>	*Boolean* 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 with 3-D Secure and Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key	*String*	Unique identifier for a Vault profile, and used in future Vault financial

Variable Name	Type and Limits	Description
<data_key>	25-character alphanumeric	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>	*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 **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>	*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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
Cardholder Authentication	*String*	Value provided by the Moneris MPI or

Variable Name	Type and Limits	Description
Verification Value (CAVV) <cavv>	50-character alphanumeric	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
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 = V, 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
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
3DS version <threeds_version> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0
3DS server transaction ID <threeds_server_trans_id> **NOTE:** Mandatory for financial transactions using 3rd Party 3-D Secure services - obtained from the Cavv Lookup request or MPI 3DS Authentication request	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction

MCP Pre-Authorization with 3-D Secure and 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 **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>	*String* 4-character alphanumeric YYMM	Expiry date of the credit card, in YYMM format. This is the reverse of the MMYY date format that is presented on the card.
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
DS transaction ID <ds_trans_id> **NOTE:** Only used in financial transactions using 3rd Party 3-D Secure services.	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.

7.11 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	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>	*Boolean* true/false	Checks whether a previously sent transaction was processed successfully To send a status check request, resend

Variable Name	Type and Limits	Description
		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>	*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.
transaction number <txn_number>	*String* 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 Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion
electronic commerce indicator	*String*	Describes the category of e-commerce transaction being processed. Allow-

Variable Name	Type and Limits	Description
<crypt_type>	1-character alphanumeric	able 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 = V, 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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
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

Variable Name	Type and Limits	Description
		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
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

7.12 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_purchasecorrection (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

Variable Name	Type and Limits	Description
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>	*Boolean* 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 Purchase Correction 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.

Variable Name	Type and Limits	Description
		For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
transaction number <txn_number>	*String* 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 Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion
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:

Variable Name	Type and Limits	Description
		if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center

7.13 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>	*Boolean* 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 Refund transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alphanumeric a-Z A-Z 0-9 _ - : . @ spaces	Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
transaction number <txn_number>	*String* 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 Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion
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

Variable Name	Type and Limits	Description
		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 = V, 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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
MCP version number <mcp_version>	*String* numeric current version is 1.0	Release version number for MCP
cardholder amount	*String*	Amount, in units of foreign currency, the cardholder will be charged on the

Variable Name	Type and Limits	Description
<cardholder_amount>	12-character numeric smallest discrete unit of foreign currency	transaction
cardholder currency code <cardholder_currency_code>	*String* 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>	*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 requests in order to redeem that rate

7.14 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	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>	*Boolean* 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,

Variable Name	Type and Limits	Description
		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>	*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 **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>	*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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 4-character alphanumeric	Expiry date of the credit card, in YYMM format.

Variable Name	Type and Limits	Description
	YYMM	**NOTE:** This is the reverse of the MMYY 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	*String*	Token representing a temporarily locked-in foreign exchange rate,

Variable Name	Type and Limits	Description
<mcp_rate_token>	N/A	obtained in the response of the MCP Get Rate transaction and used in subsequent MCP financial transaction requests in order to redeem that rate

7.15 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	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/

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>	*Boolean* 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 Purchase With Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
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

Variable Name	Type and Limits	Description
		must be the same as that of the original transaction.
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
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 = V, 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
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 Purchase 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 **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>	*String* 4-character alphanumeric	Expiry date of the credit card, in YYMM format.

Variable Name	Type and Limits	Description
	YYMM	**NOTE:** This is the reverse of the MMYY date format that is presented on the card.
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

7.16 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	Unique alphanumeric string assigned by Moneris upon merchant account activation To find your API token, refer to your

Variable Name	Type and Limits	Description
		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>	*Boolean* 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 With Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
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.
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
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

Variable Name	Type and Limits	Description
		if payment indicator = V, 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
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 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 **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	*String*	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 MMYY date format that is presented on the card.
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

7.17 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 <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

Variable Name	Type and Limits	Description
		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>	*Boolean* 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 with Vault transaction request fields – Required

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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
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
		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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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 = V, then allowable values for e-commerce indicator: 2, 5 or 6

Variable Name	Type and Limits	Description
		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>	*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 **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
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

7.18 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 <status_check>	*Boolean* 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 Get Rate transaction request fields – Required

Variable Name	Type and Limits	Description
MCP version number <mcp_version>	*String* numeric current version is 1.0	Release version number for MCP
rate transaction type <rate_txn_type>	*String* 1-character alphabetic	Value representing the type of subsequent transaction request that the rate token will be used for. Allowable values: P – Purchase R – Refund
MCP Rate Info <rate_info>	*Object* N/A	Nested object in the MCP Get Rate transaction containing the **add cardholder amount** and **add merchant settlement** fields
add cardholder amount	*String* 12-character numeric, 3-character numeric	A string array representing: • the amount, in units of foreign

Variable Name	Type and Limits	Description
	(smallest discrete unit of foreign currency, currency code)	currency, the cardholder will be charged, and • the ISO currency code corresponding to the foreign currency of the cardholder
add merchant settlement amount	*String* 12-character numeric, 3-character numeric (amount in CAD pennies, currency code)	A string array representing: • the amount the merchant will receive in the transaction, in Canadian dollars • the ISO currency code corresponding to the foreign currency of the cardholder

7.19 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)

Numeric Currency Code (ISO)	Currency Name/Acronym
036	Australian Dollar (AUD)
048	Bahraini Dinar (BHD)
050	Bangladeshi Taka (BDT)
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)

Numeric Currency Code (ISO)	Currency Name/Acronym
208	Danish Krone (DKK)
214	Dominican Republic Peso
222	Salvadoran Colon (SVC)
242	Fijian Dollar (FJD)
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)

Numeric Currency Code (ISO)	Currency Name/Acronym
404	Kenyan Shilling (KES)
410	South Korean Won (KRW)
414	Kuwaiti Dinar (KWD)
418	Laotian Kip (LAK)
426	Lesotho Loti (LSL)
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)

Numeric Currency Code (ISO)	Currency Name/Acronym
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)
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)

Numeric Currency Code (ISO)	Currency Name/Acronym
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)
840	US Dollar (USD)
858	Uruguayan Peso (UYU)
860	Uzbekistani Sum (UZS)
882	Samoan 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)

Numeric Currency Code (ISO)	Currency Name/Acronym
953	CFP Franc (XPF)
967	Zambian Kwacha (ZMW)
968	Surinamese Dollar (SRD)
969	Malagasy Ariary (MGA)
971	Afghan Afghani (AFN)
972	Tajkistan Somoni (TJS)
973	Angola Kwanza (AOA)
975	Bulgarian Lev (BGN)
977	Bosnia and Herzegovina Convertible Mark (BAM)
978	Euro (EUR)
981	Georgian Lari (GEL)
985	Polish New Zloty (PLN)
986	Brazilian Real (BRL)

7.20 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.

Error Code	Description
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

8 Apple Pay Token Transactions

- 8.1 About Apple Pay Token Transactions
- 8.2 Apple Pay Token Request DTD
- 8.3 Apple Pay Token Purchase
- 8.4 Apple Pay Token Pre-Authorization

8.1 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	`paymentAuthorizationViewController:didAuthorizePayment:handler:`
Swift	`paymentAuthorizationViewController (_:didAuthorizePayment:handler:)`
JavaScript	`ApplePaySession.onpaymentauthorized`

8.2 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)>
```

8.3 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 <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
display name	*String*	Field returned by Apple that displays

Variable Name	Type and Limits	Description
<displayName>	N/A	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: • 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: • 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
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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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

Variable Name	Type and Limits	Description
		Field is case sensitive Possible values: 3DSecure = Cryptogram obtained using MerchantCapability3DS EMV = Cryptogram obtained using MerchantCapablitiyEMV
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
token originator <token_originator> **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	*Object* 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	*String*	Unique identifier provided by Moneris

Variable Name	Type and Limits	Description
<store_id>	N/A	upon merchant account setup
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/

8.4 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>	*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

Variable Name	Type and Limits	Description
		must be the same as that of the original transaction.
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
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: • 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

Variable Name	Type and Limits	Description
		payment There are three items in the setter: • 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
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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center
network <network>	*String*	This field is mandatory for Apple Pay and Google Pay™ INTERAC® e-Com-

Variable Name	Type and Limits	Description
	N/A	merce 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 MerchantCapablitiyEMV
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 **For Pre-Authorization transactions**: the value in the dynamic descriptor field will only be carried

Variable Name	Type and Limits	Description
		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
token originator <token_originator> **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	*Object* 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>	*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/

9 Google Pay Transactions

- 9.1 About Google Pay Transactions
- 9.3 Google Pay Purchase
- 9.4 Google Pay Pre-Authorization

9.1 About Google Pay Transactions

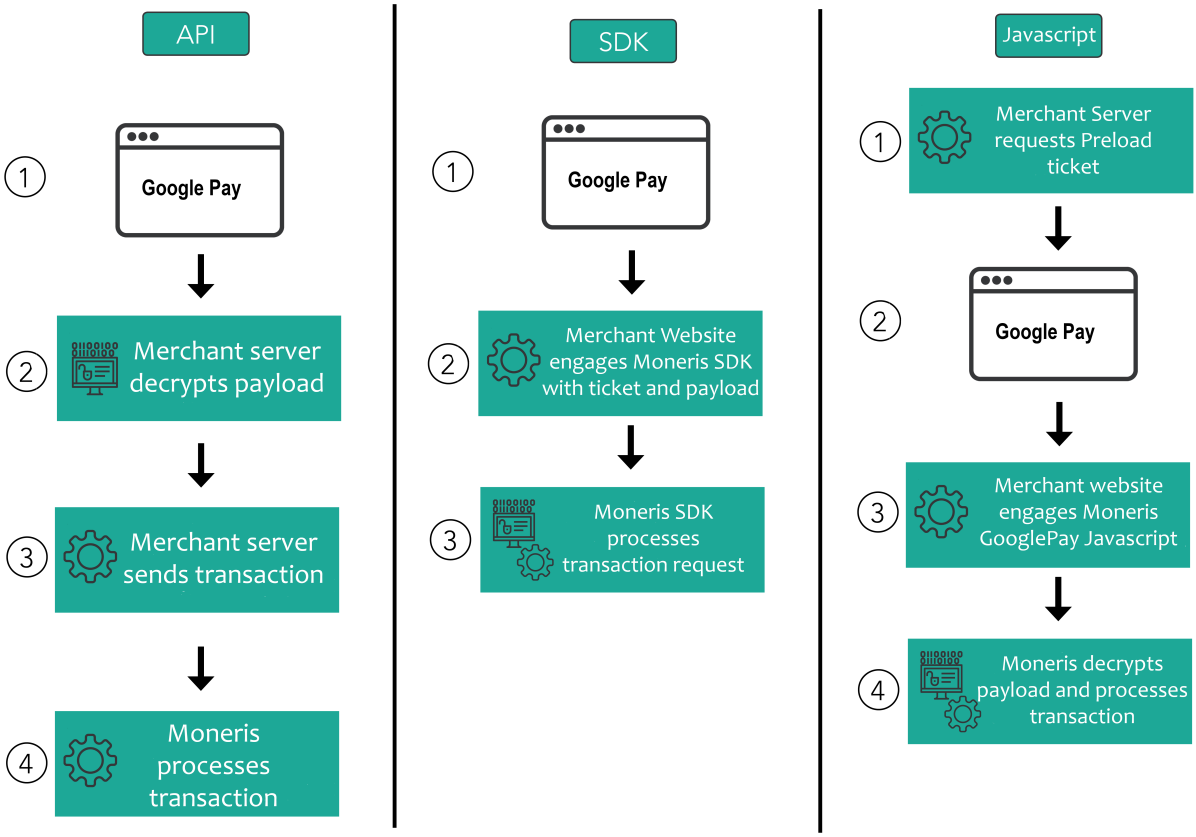
Google Pay™ transactions can be processed using the Moneris API.

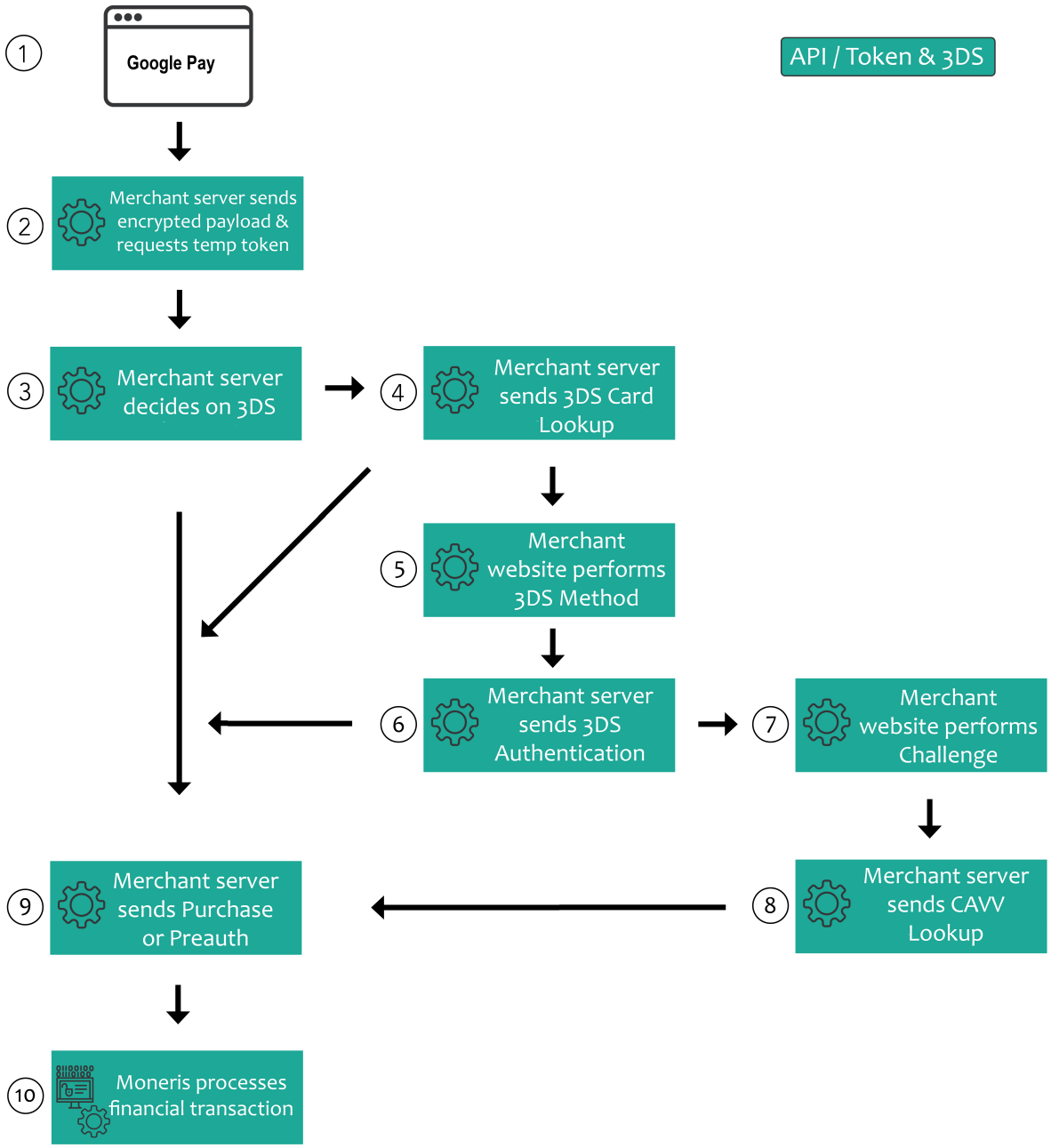
9.2 Google Pay™ Transaction Process Overview

Moneris offers four integration methods for processing transactions with Google Pay wallets. All integrations use the Google Pay™ Framework to request and receive encrypted payment details from Google. When payment details are returned in their encrypted form, the merchant can decrypt the payload on their server or transmit the encrypted payload to Moneris for decryption.

Moneris recommends merchants utilize the 3DS process on Google Pay temporary tokens with an underlying card type of FPAN to reduce risk of fraud and chargebacks. Merchants may attempt or skip the 3DS process if the underlying card type is DPAN; the card issuer may not support 3DS for the device PAN, however, so Moneris recommends using the 3DS Card Lookup to ensure support for 3DS.

Google Pay™ Integration Methods & Process


API
1
Google Pay
2
Merchant server decrypts payload
3
Merchant server sends transaction
4
Moneris processes transaction
SDK
1
Google Pay
2
Merchant Website engages Moneris SDK with ticket and payload
3
Moneris SDK processes transaction request
Javascript
1
Merchant Server requests Preload ticket
2
Google Pay
3
Merchant website engages Moneris GooglePay Javascript
4
Moneris decrypts payload and processes transaction


1
Google Pay
API / Token & 3DS
2
Merchant server sends encrypted payload & requests temp token
3
Merchant server decides on 3DS
4
Merchant server sends 3DS Card Lookup
5
Merchant website performs 3DS Method
6
Merchant server sends 3DS Authentication
7
Merchant website performs Challenge
8
Merchant server sends CAVV Lookup
9
Merchant server sends Purchase or Preauth
10
Moneris processes financial transaction

Decrypted Payload (Merchant server to Moneris Gateway API)

Merchant decrypts the Google Pay encrypted payload locally then processes a standard financial transaction via Moneris Gateway API call with card data. Used by both in-app and web solutions.

> **NOTE:** In this API scenario where merchant's server is responsible for decrypting the payload, merchants must sign agreement with Google directly. Google can then provide you with the keys to decrypt the payload.

1. Merchant's mobile application or web page requests and receives the encrypted payload from Google

2. Encrypted payload is sent from the merchant's website to Moneris Gateway via the SDK, and the payload is decrypted and processed

Encrypted Payload (Merchant website to Moneris SDK)

Merchant passes the encrypted payload to the Moneris Google Pay SDK. Used for in-app solutions only. The SDK files are located on the Moneris Github and instructions on integration found on our Moneris Developer Portal.

1. Merchant's mobile application or web page requests and receives the encrypted payload from Google

2. Encrypted payload is sent from the merchant's website to Moneris Gateway via the SDK, and the payload is decrypted and processed

Encrypted Payload (Merchant website to Moneris JavaScript)

Merchant edits the Google Pay Javascript to utilize Moneris Google Pay for processing the payment. Includes an optional Preload via Moneris Gateway API specific to this integration method. See the full guide located on our Moneris Github or instructions on our Moneris Developer Portal.

1. Merchant's server populates a Preload request and receives a ticket (optional step)

2. Merchant's mobile application or web page requests and receives the encrypted payload from Google

3. Encrypted payload is sent from the merchant's application or website to the via embedded Moneris Javascript, and the payload is decrypted and processed.

Encrypted Payload With 3DS (Merchant server to Moneris Gateway API)

Merchant transmits the encrypted payload via Moneris Gateway API to decrypt and tokenize the card data temporarily. This temporary token is usable for performing 3D-Secure authentication and the subsequent financial transaction.

> **NOTE:** Moneris recommends merchants utilize the 3DS process on Google Pay temporary tokens with an underlying card type of FPAN to reduce risk of fraud and chargebacks. Merchants may attempt or skip the 3DS process if the underlying card type is DPAN; the card issuer may not support 3DS for the device PAN, however, so Moneris recommends using the 3DS Card Lookup to ensure support for 3DS.

1. Merchant's app or web page requests and receives the encrypted payload from Google

2. Encrypted payload is sent from the merchant's server to the via a GooglePay Temporary Token Add. Moneris returns a temporay payment token in the response and a GooglePayPaymentMethod indicating the type of underlying card data (FPAN or DPAN)

3. Merchant server elects whether to perform 3DS Authentication or not.If electing to skip 3DS Authentication, the merchant server can skip to Step 9 and immediately perform a financial transaction.

4. Merchant server sends the temporary token in a 3DS Card Lookup request. Moneris responds with whether the underlying card supports 3DS authentication or not and details for the 3DS Method, if available.

   For cards that do not support 3DS, the merchant server should skip the rest of the 3DS Authentication flow and move to Step 9.

5. If available, merchant server and app performs the 3DS Method using the 3DSMethodData and 3DSMethodURL. See "Handling the 3DS Method for Device Fingerprinting" on page 133

6. Merchant server performs a 3DS Authentication request (Browser Channel) to the Moneris Gateway. See "Implementing MPI 3DS Authentication Request" on page 133

   If Moneris responds with a successful result (frictionless), the merchant server receives the CAVV and ECI values from the 3DS authentication response itself. Skip the challenge flow and move to Step 9.

   If Moneris responds that a challenge prompt (friction) is required, continue with the next step.

7. Merchant server and application proceed with the 3DS Challenge. See "Handling the Challenge Flow" on page 164

8. Merchant server sends CAVV Lookup request to retrieve the authentication value (CAVV) and ecommerce indicator (crypt_type) after the challenge is completed.

9. Merchant server performs either a GooglePayTokenPreauth or GooglePayTokenPurchase as the financial transaction. The CAVV and electronic commerice indicator (crypt_type) are included as follows:

   If 3DS was skipped earlier, omit the CAVV field and use the appropriate ECI for your transaction type.

   If 3DS authentication was performed successfully, supply the CAVV from your 3DS Authentication or CAVV Lookup response.

9.3 Google Pay Purchase

Purchase transaction using Google Pay™

XML transaction object

```
<googlepay_purchase>
```

Google Pay Purchase transaction object definition

```
<!ELEMENT googlepay_purchase (order_id, cust_id?, amount, network, payment_
token, 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	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>	*Boolean* 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

Google Pay 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 <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
network <network>	*String* alphabetic	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
payment token <payment_token>	*String* 32-character alphanumeric	Payment token submitted by merchant for order (credit card, payer ID, routing/transit, MICR, and account number) If payment_type is set to None then the payment_token value should be left empty (NULL) If the credit card information is not available and a Moneris Vault token is used to process payment set payment_type = TOKEN and send the token (data key) in the payment_token field.

Google Pay Purchase 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 **NOTE:** The 22-character maximum limit must take the "/" into account as one of

Variable Name	Type and Limits	Description
		the characters

9.4 Google Pay Pre-Authorization

Pre-Authorization transaction using Google Pay™.

XML transaction object

```
<googlepay_preauth>
```

Google Pay Pre-Authorization transaction object definition

```
<!ELEMENT googlepay_preauth (order_id, cust_id?, amount, network, payment_
token, 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	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>	*Boolean* true/false	Checks whether a previously sent transaction was processed successfully

Variable Name	Type and Limits	Description
		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

Google Pay 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 have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
network <network>	*String* alphabetic	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.

Variable Name	Type and Limits	Description
		Field is case sensitive Possible value: Interac
payment token <payment_token>	*String* 32-character alphanumeric	Payment token submitted by merchant for order (credit card, payer ID, routing/transit, MICR, and account number) If payment_type is set to None then the payment_token value should be left empty (NULL) If the credit card information is not available and a Moneris Vault token is used to process payment set payment_type = TOKEN and send the token (data key) in the payment_ token field.

Google Pay Pre-Authorization 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

Variable Name	Type and Limits	Description
	< > $ % = ? ^ { } [ ] \	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

9.5 Google Pay Temporary Token Add

Creates a new temporary token credit card profile from an encrypted GooglePay payload. During the lifetime of this temporary token, it may be used to perform 3DS authentication and financial transactions via GooglePay Token Preauthorization or GooglePay Token Purchase.

The response field `GooglePaymentMethod` returned by this request will inform you if the underlying card within GooglePay is the funding card number ("FPAN") or a tokenized card number ("DPAN"). If a GoogleTokenTempAdd returns an FPAN, you may peform 3DS authentication with it; if it returns a DPAN, 3DS is not required.

Refer to Apple or Google developer portals for details on integrating directly to their wallets to retrieve the payload data.

Things to Consider:

- The duration, or lifetime, of the temporary token can be set to be a maximum of 15 minutes.

XML transaction object

```
<googlepay_token_temp_add>
```

Google Pay Purchase transaction object definition

```
<!ELEMENT googlepay_token_temp_add (order_id?, cust_id?, network, payment_
token, dynamic_descriptor?)>

<!ELEMENT payment_token (signature, protocol_version, signed_message)>
```

Core connection object fields (all API transactions)

Variable Name	Type and Limits	Description
store ID	*String*	Unique identifier provided by Moneris

Variable Name	Type and Limits	Description
<store_id>	N/A	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>	*Boolean* 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

Google Pay Temporary Token Add 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
		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.
network <network>	*String* alphabetic	Card Brand name. Field is case sensitive Possible values: Visa Mastercard American Express Interac Discover
payment token <payment_token>	*String* 32-character alphanumeric	Payment details returned by Google in their `PaymentData` object for GooglePay transactions. See GooglePay Payment Token object request fields – Required below for field details.

Google Pay Temporary Token Add 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 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 **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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

GooglePay Payment Token object request fields – Required

Variable Name	Type and Limits	Set Method
signature	*String* N/A	Verifies that the message came from Google. It's base64-encoded, and created with ECDSA by the intermediate signing key. Returned by Google in their `PaymentData` object for GooglePay transactions
protocol version	*String* N/A	Identifies the encryption or signing scheme under which the message is created. It allows the protocol to evolve over time, if needed. Returned by Google in their `PaymentData` object for GooglePay transactions
signed message	*String* N/A	A JSON object serialized as an HTML-safe string that contains the encryptedMessage, ephemeralPublicKey, and tag. It's serialized to simplify the signature verification process. Returned by Google in their `PaymentData` object for GooglePay transactions

SampleGoogle Pay™ Temporary Token Add

```
Request:
<?xml version="1.0" encoding="UTF-8"?>
```

```
<request>
  <store_id>store5</store_id>
  <api_token>yesguy</api_token>
  <googlepay_token_temp_add>
    <network>VISA</network>
    <payment_token>
      <signature>MEUCIA6jKuw...I+LHnbRIgdsrKgOAETqHhNhNK6LmM=</signature>
      <protocol_version>ECv1</protocol_version>
      <signed_message>
{"encryptedMessage":"094Co6Xv3Bwjc1JH9MSlEXDnSE3e8+FRmZXY8RdxHYCTiF1fOSnngaCmiYLK54EeKcEASzh/3Y98
wdELGzWIHgf3usn2aqPjcvoS5iJSC/vq8Vl+tVmb650kS4N/QphyN+WDdvbHnUuM5hgbXJ+jxT8XL01fpqpB848YgT4O+xyLA
RUGIEfM++V7X4z/wlOXWDO1ZAnPF5ndPUSnxRN3aWSz22u3fSYBiKaZtGwbDdnjw5XjbYYiotLkcCqItjRM+shIzvrF/8qaY5
Z+pV1Rdaro4gJNQgjGdINJ1DSj9PV2cdGJhYgTzuCiu6k5UgADh4lKcUfF3+tzZOA7fJ+2nG85Vi6CFpYKGBctprnJhf7axOn
QS+xAODauQSBWNPELZ0oJVyuB9xpBOvkxv25Th33R8giLd2zM1CJpKg2aw/2yQxjx4AH2Nu/T9HnFQqZEpw\u003d\u003d",
"ephemeralPublicKey":"BCCUC1P9WJDU96Hr50d5GBON6CaGjr2Xb4ZA3ueFJXhzv5RMcHpt52RuHedsZraLWCwiOwXhnoK
znYAuqsHykOQ\u003d","tag":"nWd0IZKCNrmB++b3h5M5WSAg/zjURNBOqb8iIK18OBo\u003d"}</signed_message>
    </payment_token>
  </googlepay_token_temp_add>
</request>
```

Response:

```
<?xml version="1.0"?>
<response>
       <receipt>
              <DataKey>ot-R1jWBQGcM1kJBHDyFrcSG85c2</DataKey>
              <ReceiptId>null</ReceiptId>
              <ReferenceNum>null</ReferenceNum>
              <ResponseCode>001</ResponseCode>
              <ISO>null</ISO>
              <AuthCode>null</AuthCode>
              <Message>Successfully registered CC details.</Message>
              <TransTime>16:41:54</TransTime>
              <TransDate>2023/11/23</TransDate>
              <TransType>null</TransType>
              <Complete>true</Complete>
              <TransAmount>null</TransAmount>
              <CardType>null</CardType>
              <TransID>null</TransID>
              <TimedOut>false</TimedOut>
              <CorporateCard>null</CorporateCard>
              <RecurSuccess>null</RecurSuccess>
              <AvsResultCode>null</AvsResultCode>
```

```
                <CvdResultCode>null</CvdResultCode>
                <ResSuccess>true</ResSuccess>
                <PaymentType>cc</PaymentType>
                <IsVisaDebit>null</IsVisaDebit>
                <ResolveData>
                        <anc1></anc1>
                        <masked_pan>4242***4242</masked_pan>
                        <expdate>2305</expdate>
                </ResolveData>
                <GooglepayPaymentMethod>FPAN/DPAN</GooglepayPaymentMethod>
        </receipt>
</response>
```

9.6 Google Pay Token Purchase

The Google Pay™ Token Purchase transaction is utilized after passing a GooglePay account into a temporary token using our GooglePay Token Temporary Add then performing 3DS authentication with the token. This transaction verifies funds on the customer's card, removes the funds and prepares them for deposit into the merchant's account.

To perform the 3-D Secure authentication, the Moneris MPI or any third-party MPI may be used.

Refer to Apple or Google developer portals for details on integrating directly to their wallets to retrieve the payload data.

XML transaction object

```
<googlepay_token_purchase>
```

Google Pay Purchase transaction object definition

```
<!ELEMENT googlepay_token_purchase (order_id, amount, data_key, crypt_type,
cavv, threeds_server_trans_id?, threeds_version?, ds_trans_id?, 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	Unique alphanumeric string assigned by Moneris upon merchant account activation

Variable Name	Type and Limits	Description
		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>	*Boolean* 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

Google Pay Temporary Token Add 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.

Variable Name	Type and Limits	Description
data key <data_key>	*String* 25-character alphanumeric	The temporary token returned by a `GooglePayTokenTempAdd` request.
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
Cardholder Authentication Verification Value (CAVV) <cavv> **NOTE:** For Google Pay™ Token Purchase transactions, CAVV field contains the 3DS cryptogram only when 3DS is used prior. If you elected to skip 3DS Authentication, you may omit the CAVV field.	*String* 50-character alphanumeric	The 3DS cryptogram. Sent in all financial transactions with 3-D Secure, including Verified By Visa, MasterCard SecureCode, American Express SafeKey
electronic commerce indicator <crypt_type> **NOTE:** For Google Pay™ Token Purchase and Token Pre-Authorization transactions using 3DS Authentication, use the e-commerce indicator obtained from your 3DS Authentication.	*String* 1-character alphanumeric	
3DS server transaction ID	*String*	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication

Variable Name	Type and Limits	Description
<threeds_server_trans_id> **NOTE:** For Google Pay™ Token Purchase and Token Pre-Authorization transactions that do not use 3DS Authentication, you may omit the 3DS Server Transaction ID.	36-character numeric	Request transaction
3DS version <threeds_version> **NOTE:** If you elected to skip 3DS Authentication, you may omit the 3DS Version field.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0

Google Pay Temporary Token Add 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

Variable Name	Type and Limits	Description
	< > $ % = ? ^ { } [ ] \	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

SampleGoogle Pay™ Token Purchase

```
Request:
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <store_id>monca03035</store_id>
  <api_token>qnTUYAL2yR6KDxVlSLKa</api_token>
  <googlepay_token_purchase>
        <order_id>Test1708445437248</order_id>
        <cust_id>nqa-cust_id</cust_id>
        <amount>1.00</amount>
        <dynamic_descriptor>nqa-dd</dynamic_descriptor>
        <crypt_type>2</crypt_type>
        <data_key>ot-cEOTTMEuEGzcMjftc26ImZbw1</data_key>
        <threeds_server_trans_id>de1b97ee-c610-4877-b53f-c1c5ecd99bf0</threeds_server_trans_id>
        <ds_trans_id>de1b97ee-c610-4877-b53f-c1c5ecd99bf0</ds_trans_id>
        <threeds_version>2.2</threeds_version>
        <cavv>kAABApFSYyd4l2eQQFJjAAAAAAA=</cavv>
    </googlepay_token_ purchase>
</request>


Response:
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <receipt>
        <ReceiptId>Test1708445437248</ReceiptId>
        <ReferenceNum>660183980017560080</ReferenceNum>
        <ResponseCode>027</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>KN7300</AuthCode>
        <TransTime>16:52:09</TransTime>
        <TransDate>2024-01-18</TransDate>
        <TransType>00</TransType>
```

```
        <Complete>true</Complete>
        <Message>APPROVED * =</Message>
        <TransAmount>8.00</TransAmount>
        <CardType>V</CardType>
        <TransID>31730-0_558</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
        <CavvResultCode>2</CavvResultCode>
        <Par>4761AB123456789C1231111111111</Par>
        <IsVisaDebit>false</IsVisaDebit>
        <ThreeDSVersion>null</ThreeDSVersion>
        <GooglepayPaymentMethod>FPAN</GooglepayPaymentMethod>
    </receipt>
</response>
```

9.7 Google Pay Token Preauth

The Google Pay™ Token Preauth transaction is utilized after passing a GooglePay account into a temporary token using our GooglePay Token Temporary Add then performing 3DS authentication with the token. This transaction verifies funds on the customer's card and locks those funds for a time period specified by the card issuer.

To perform the 3-D Secure authentication, the Moneris MPI or any third-party MPI may be used.

Refer to Apple or Google developer portals for details on integrating directly to their wallets to retrieve the payload data.

XML transaction object

```
<googlepay_token_preauth>
```

Google Pay Preauth transaction object definition

```
<!ELEMENT googlepay_token_preauth (order_id, amount, data_key, crypt_type,
cavv, threeds_server_trans_id?, threeds_version?, ds_trans_id?, 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	*String*	Unique alphanumeric string assigned

Variable Name	Type and Limits	Description
<api_token>	N/A	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>	*Boolean* 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

Google Pay Temporary Token Add 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

Variable Name	Type and Limits	Description
		Correction transactions, the order ID must be the same as that of the original transaction.
data key <data_key>	*String* 25-character alphanumeric	The temporary token returned by a `GooglePayTokenTempAdd` request.
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
Cardholder Authentication Verification Value (CAVV) <cavv> **NOTE:** For Google Pay™ Token Purchase transactions, CAVV field contains the 3DS cryptogram only when 3DS is used prior. If you elected to skip 3DS Authentication, you may omit the CAVV field.	*String* 50-character alphanumeric	The 3DS cryptogram. Sent in all financial transactions with 3-D Secure, including Verified By Visa, MasterCard SecureCode, American Express SafeKey
electronic commerce indicator <crypt_type> **NOTE:** For Google Pay™ Token Purchase and Token Pre-Authorization transactions using 3DS Authentication, use the e-commerce indicator obtained from your 3DS	*String* 1-character alphanumeric	

Variable Name	Type and Limits	Description
Authentication.		
3DS server transaction ID <threeds_server_trans_id> **NOTE:** For Google Pay™ Token Purchase and Token Pre-Authorization transactions that do not use 3DS Authentication, you may omit the 3DS Server Transaction ID.	*String* 36-character numeric	Data is obtained from a Cavv Lookup Request or MPI 3DS Authentication Request transaction
3DS version <threeds_version> **NOTE:** If you elected to skip 3DS Authentication, you may omit the 3DS Version field.	*String* 10-character numeric	Acceptable values: 2.0.0 = 3DS protocol 2.0.0 2.1.0 = 3DS protocol 2.1.0 2.2.0 = 3DS protocol 2.2.0 2.3.0 = 3DS protocol 2.3.0

Google Pay Temporary Token Add 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	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name

Variable Name	Type and Limits	Description
	and separator **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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

SampleGoogle Pay™ Token Preauth

```
Request:
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <store_id>monca03035</store_id>
  <api_token>qnTUYAL2yR6KDxVlSLKa</api_token>
  <googlepay_token_preauth>
        <order_id>Test1708445437248</order_id>
        <cust_id>nqa-cust_id</cust_id>
        <amount>1.00</amount>
        <dynamic_descriptor>nqa-dd</dynamic_descriptor>
        <crypt_type>2</crypt_type>
        <data_key>ot-cEOTTMEuEGzcMjftc26ImZbw1</data_key>
        <threeds_server_trans_id>de1b97ee-c610-4877-b53f-c1c5ecd99bf0</threeds_server_trans_id>
        <ds_trans_id>de1b97ee-c610-4877-b53f-c1c5ecd99bf0</ds_trans_id>
        <threeds_version>2.2</threeds_version>
        <cavv>kAABApFSYyd4l2eQQFJjAAAAAAA=</cavv>
    </googlepay_token_preauth>
</request>

Response:
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <receipt>
        <ReceiptId>Test1708445437248</ReceiptId>
        <ReferenceNum>660183980017560080</ReferenceNum>
        <ResponseCode>027</ResponseCode>
        <ISO>01</ISO>
        <AuthCode>KN7300</AuthCode>
```

```
        <TransTime>16:52:09</TransTime>
        <TransDate>2024-01-18</TransDate>
        <TransType>00</TransType>
        <Complete>true</Complete>
        <Message>APPROVED * =</Message>
        <TransAmount>8.00</TransAmount>
        <CardType>V</CardType>
        <TransID>31730-0_558</TransID>
        <TimedOut>false</TimedOut>
        <BankTotals>null</BankTotals>
        <Ticket>null</Ticket>
        <CavvResultCode>2</CavvResultCode>
        <Par>4761AB123456789C1231111111111</Par>
        <IsVisaDebit>false</IsVisaDebit>
        <ThreeDSVersion>null</ThreeDSVersion>
        <GooglepayPaymentMethod>FPAN</GooglepayPaymentMethod>
    </receipt>
</response>
```

10 Recurring Billing

- 10.1 About Recurring Billing
- 10.2 Purchase with Recurring Billing
- 10.3 Recurring Billing Update
- 10.4 Recurring Billing Response Fields and Codes
- 10.5 Credential on File and Recurring Billing

10.1 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.

10.2 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?, PBB_
info?, installment_info?, wallet_indicator?, foreign_indicator?)>
```

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>	*Boolean* 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

Purchase with Recurring Billing transaction request fields – Required

Variable Name	Type and Limits	Description
order ID	*String*	Merchant-defined transaction identifier that must be unique for every

Variable Name	Type and Limits	Description
<order_id>	50-character alpha-numerica-Z A-Z 0-9 _ - : . @ spaces	Purchase, Pre-Authorization and Independent Refund transaction. No two transactions of these types may have the same order ID. For Refund, Completion and Purchase Correction transactions, the order ID must be the same as that of the original transaction.
amount <amount>	*String* 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
credit card number <pan >	*String* 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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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

Variable Name	Type and Limits	Description
		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 = V, 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 Recurring Billing 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	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement

Variable Name	Type and Limits	Description
	total of 22 characters including your merchant name and separator **NOTE:** Some special characters are not allowed: < > $ % = ? ^ { } [ ] \	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
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not sup-

Variable Name	Type and Limits	Description
		ported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).
Customer Information <cust_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
AVS Information <avs_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
CVD Information <cvd_info> For information on request fields for this object, see xrefHere	*Object* N/A	Contains fields related to the Card Validation Digits e-fraud tool
Recurring Billing <recur> For information on request fields for this object, see xrefHere	*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.

Variable Name	Type and Limits	Description
For information on request fields for this object, see xre-fHere		

Recurring Billing object request fields

Variable Name	Type and Limits	Description
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 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
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
number of recurs <num_recurs>	*String* numeric 1-999	The number of times that the transaction must recur
period <period>	*String* numeric	Number of recur unit intervals that must pass between recurring billings

Variable Name	Type and Limits	Description
	1-999	
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 **EXAMPLE:** 1234567.89	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

10.3 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>	*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
		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>	*Boolean* 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

Recurring Billing Update 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

Variable Name	Type and Limits	Description
		that of the original transaction.

Recurring Billing Update 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
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. Carries the token for network tokenization transactions.
expiry date <expdate>	*String* 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.
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 **EXAMPLE:** 1234567.89	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	*String*	Increments a specified number of

Variable Name	Type and Limits	Description
<add_num_recurs>	3-character numeric 1-999	transactions to the current remaining 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_recurs>	*String* 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>	*Boolean* 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>	*Boolean* 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>	*Object* N/A	Required when storing cardholder credentials or using these credentials in subsequent transactions.

10.4 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

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	
	See Table 3: for a description of possible response codes.		
Recur success	String	TBD	
	Indicates whether the transaction successfully registered		
Recur update object response fields			
Recur update success	String	`true`/`false`	
	Indicates whether the transaction successfully updated.		
Next recur date	String	yyyy-mm-dd format	
	Indicates when the transaction will be billed again.		
Recur end date	String	yyyy-mm-dd format	
	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

10.5 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

11 Customer Information

- 11.1 Customer Information Object
- 11.2 Customer Info Object – Billing Information
- 11.3 Customer Info Object – Shipping Information
- 11.4 Customer Information Object – Items

11.1 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>	*Object* N/A	Sub-object of the customer information object; contains fields related to shipping
shipping information <shipping>	*Object* N/A	Sub-object of the customer information object; contains fields related to shipping
email <email>	*String* 60-character alphanumeric	Customer's email address
instructions <instructions>	*String* 100-character alphanumeric	Instructions or notes

11.2 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>	*String* 30-character alphanumeric	Customer first name
last name <last_name>	*String* 30-character alphanumeric	Customer last name
company name <company_name>	*String* 50-character alphanumeric	Customer's company name
address	*String*	Customer address

Variable Name	Type and Limits	Description
<address>	70-character alphanumeric	
city <city>	*String* 30-character alphanumeric	Customer city
province/state <province>	*String* 30-character alphanumeric	Customer province or state
postal/ZIP code <postal_code>	*String* 30-character alphanumeric	Customer postal or ZIP code
country <country>	*String* 30-character alphanumeric	Customer's country
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number
fax number <fax>	*String* 30-character alphanumeric	Customer fax number
federal tax <tax1>	*String* 10-character alphanumeric	Dollar amount of federal tax Not used to calculate total amount
provincial/state tax <tax2>	*String* 10-character alphanumeric	Dollar amount of provincial or state tax Not used to calculate total amount
county/local/specialty tax <tax3>	*String* 10-character alphanumeric	Dollar amount of county, local or specialty tax amount Not used to calculate total amount
shipping cost <shipping_cost>	*String* 10-character alphanumeric	Dollar amount of fees charged for shipping Not used to calculate total amount

11.3 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>	*String* 30-character alphanumeric	Customer first name
last name <last_name>	*String* 30-character alphanumeric	Customer last name
company name <company_name>	*String* 50-character alphanumeric	Sub-object of the customer information object; contains fields related to shipping
address <address>	*String* 70-character alphanumeric	Customer address
city <city>	*String* 30-character alphanumeric	Customer city
province/state <province>	*String* 30-character alphanumeric	Customer province or state
postal/ZIP code <postal_code>	*String* 30-character alphanumeric	Customer postal or ZIP code
country <country>	*String* 30-character alphanumeric	Customer's country
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number
fax number <fax>	*String* 30-character alphanumeric	Customer fax number

Variable Name	Type and Limits	Description
federal tax <tax1>	*String* 10-character alphanumeric	Dollar amount of federal tax Not used to calculate total amount
province/state <tax2>	*String* 30-character alphanumeric	Customer province or state
county/local/specialty tax <tax3>	*String* 10-character alphanumeric	Dollar amount of county, local or specialty tax amount Not used to calculate total amount
shipping cost <shipping_cost>	*String* 10-character alphanumeric	Dollar amount of fees charged for shipping Not used to calculate total amount

11.4 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>	*String* 45-character alphanumeric	Name of a specific item being purchased
item quantity <quantity>	*String* 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>	*String* 20-character alphanumeric	Product code or SKU of an item being purchased
item extended amount <extended_amount>	*String* 10-character decimal; must contain minimum 3 digits	Unit cost, in dollars, multiplied by quantity ordered

Variable Name	Type and Limits	Description
	and 2 penny values Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point **EXAMPLE:** 1234567.89	

12 e-Fraud Tools

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

12.1 Address Verification Service (AVS)

- 12.1.1 About Address Verification Service (AVS)
- 12.1.2 AVS Information Object
- 12.1.3 AVS Response Codes

12.1.1 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.

12.1.2 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_number>	*String* 19-character alphanumeric	Cardholder's address street number
AVS street name <avs_street_name>	*String* 19-character alphanumeric	Cardholder's address street name
AVS postal/ZIP code <avs_zipcode>	*String* 9-character alphanumeric	Cardholder's address postal or ZIP code

12.1.3 AVS Response Codes

Below is a full list of possible AVS response codes.

Code	Visa	Mastercard/Discover	American Express/ JCB
A	AVS street address only (partial match)	Address matches, zip/ postal code does not	Billing address matches, zip/postal code does not
D	No longer applicable to Visa	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	No longer applicable to Visa	N/A	Customer name incorrect; billing address matches
G	No longer applicable to Visa	Address information not verified for international transaction	N/A
K	N/A	N/A	Customer name matches
L	N/A	N/A	Customer name and zip/postal code match
M	No longer applicable to Visa	N/A	Customer name, billing address, and zip/postal code match
N	AVS non-match	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

Code	Visa	Mastercard/Discover	American Express/ JCB
R	AVS indeterminate outcome (retry)	Retry; system unable to process	System unavailable; retry
S	No longer applicable to Visa	AVS currently not supported	AVS currently not supported
T	N/A	Nine-digit zip code matches; address does not match	N/A
U	AVS unable to verify	No data from issuer-/authorization system	Information is unavailable
W	No longer applicable to Visa	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	No longer applicable to Visa	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	AVS full match	Billing address and zip/postal code both match	Billing address and zip/-postal code both match
Z	AVS zip/postal code only (partial match)	For U.S. addresses, five-digit zip code matches, address does not match	Zip/postal code matches, billing address does not

12.2 Card Validation Digits (CVD)

- 12.2.1 About Card Validation Digits (CVD)
- 12.2.2 Transactions Where CVD Is Required

- 12.2.3 CVD Information Object
- 12.2.4 CVD Result Codes

12.2.1 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.

12.2.2 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

12.2.3 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.

12.2.4 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
Other	Invalid response code

Appendix A 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 amountamount <comp_amount>	*String* 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point	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
	EXAMPLE: 1234567.89	
credit card number <pan>	*String* 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. Carries the token for network token-ization transactions.
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 Mer-chant Resource Center
dynamic descriptor <dynamic_descriptor>	*String* 20-character alphanumeric total of 22 characters includ-ing 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 busi-ness 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; addi-tional characters will be truncated **NOTE:** The 22-character maximum limit must take the "/" into account as one of the characters
electronic commerce indic-ator <crypt_type>	*String* 1-character alphanumeric	Describes the category of e-com-merce transaction being processed. Allowable values are: 1 – Mail Order / Telephone Order—Single 2 – Mail Order / Telephone Order—Recurring

Variable Name	Type and Limits	Description
		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 = V, 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
expiry date <expdate>	*String* 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.
is incremental is_incremental	*Boolean* true/false	Indicates if this preauthorization is using an estimated amount. Estimations allow for incrementing the amount held via subsequent incrementalAuth requests. Defaults to

Variable Name	Type and Limits	Description
		false. **NOTE:** Please note that if this field is true, the preauthorization is only eligible for a single Preauthorization Completion. Any completion sent for partial completion is treated as a full completion (ship_indicator= P is treated as = F when is_incremental= true on the original preauth)
foreign indicator <foreign_indicator>	*Boolean* true or false	Used to identify domestic transactions processed by a marketplace merchant that is in a different country.
order ID <order_id>	*String* 50-character alpha-numerica-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.
original order ID <orig_order_id>	*String* 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>	*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

Variable Name	Type and Limits	Description
		by the issuer as supplemental shipments or completions are pending, submit shipping indicator with a value of P Possible values: P – Partial F – Final
status check <status_check>	*Boolean* 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
store ID <store_id>	*String* N/A	Unique identifier provided by Moneris upon merchant account setup
transaction number <txn_number>	*String* 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 Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion

Variable Name	Type and Limits	Description
wallet indicator <wallet_indicator>	*String* 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, or via network tokenization from the card brand. Required for Apple Pay, 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 GPW – Google Pay™ Web VCO –Visa Checkout MMP – Mastercard MasterPass **NOTE:** Please note that if this field is included to indicate Apple Pay or Google Pay™, then Convenience Fee is not supported. **NOTE:** Network tokenization wallet indicators are not in the API call but are in the merchant resource centre (MRC).

A.1 Definition of Request Fields – Admin Transactions

Variable Name	Type and Limits	Description
electronic cash register	*String*	Identification number assigned to a

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

A.2 Definition of Request Fields – Vault

Request fields applicable to Vault transactions

Variable Name	Type and Limits	Description
data key <data_key>	*String* 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>	*String* 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>	*String* 3-character numeric maximum 900 seconds	Amount of time the temporary token should be available
email address <email>	*String* 30-character alphanumeric	Customer's email address Can be sent in when creating or updating a Vault profile
note <note>	*String* 30-character alphanumeric	Used for any supplementary information related to the customer

Variable Name	Type and Limits	Description
		Can be sent in when creating or updating a Vault profile
phone number <phone>	*String* 30-character alphanumeric	Customer's phone number Can be sent in when creating or updating a Vault profile
return issuer ID <return_issuer_id>	*Boolean* true/false	When true, Gateway returns the bank Issuer ID. Defaults to False.

A.3 Definition of Request Fields – 3-D Secure 2.2

Variable Name	Type and Limits	Description
billing address bill_address1	*String* 50-character alphanumeric	Cardholder billing address
billing city bill_city	*String* 50-character alphanumeric	Cardholder billing city
billing country bill_country	*String* 3-character alphanumeric	Defined as 3 digit country code ISO 3166-1
billing postal code bill_postal_code	*String* 16-character alphanumeric	Cardholder billing postal code
billing province bill_province	*String* 3-character alphanumeric	Cardholder province or state Defined in country subdivision ISO 3166-2
browser java enabled browser_java_enabled	*String* 1-character alphabetic	Indicates whether Java is enabled in the browser Allowable values: T = True F = False

Variable Name	Type and Limits	Description
browser language browser_language	*String* 8-character alphanumeric	As defined in IETF BCP47
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 user agent browser_user_agent	*String* 2048-character alpha-numeric	Browser User Agent
cardholder name cardholder_name	*String* 45-character alphanumeric **NOTE:** Accented characters are not allowable	Name of the cardholder
challenge window size challengeWindowsize	*String* 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
cres cres	*String* 200-character alphanumeric	Response data from the challenge
currency currency	*String* 3-character numeric	ISO 4217 3 digit currency code (CAD = 124, USD = 840) **NOTE:** This field should not be sent unless

Variable Name	Type and Limits	Description
		Multi Currency Pricing is enabled on your merchant account
DS transaction ID ds_trans_id **NOTE:** Only used in financial transactions using 3rd Party 3-D Secure services.	*String* 36-character alphanumeric	Refers to the DSTransID in the response of the previous 3DS authentication.
decoupled request async URL decoupled_request_async_url	*String* 256-character alphanumeric	Your URL where Moneris will POST the response back from ACS. Moneris reattempts 3 times to POST the response.
decoupled request indicator decoupled_request_indicator	*String* 1-character alphabetic	Whether the request utilizes Decoupled Authentication or not, if the ACS confirms its use. Y = Decoupled Authentication is supported and preferred if challenge is necessary N = Do not use Decoupled Authentication (Default)
decoupled request max time decoupled_request_max_time	*String* 5-character numeric	The maximum minutes that Moneris waits for an ACS to provide results. Numeric values between 1 and 10080. The max is equivalent to 7 days.
device channel device_channel	*String* 2-character numeric	The interface used to initiate the authentication: 02 = Browser (BRW) 03 = 3DS Requestor Initiated (3RI)
email email	*String*	Cardholder email address

Variable Name	Type and Limits	Description
	254-character alphanumeric	**NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide the cardholder's email address. Lack of providing the cardholder's address, might increase the risk of rejects.
message category message_category	*String* 2-character numeric	Whether the authentication request is for a payment or non-payment use: 01 = payment authentication (PA) 02 = non-payment authentication (NPA)
notification URL notification_url	*String* 256-character alphanumeric	Notification URL for receiving the 3DS Method POST response from the issuer ACS.
prior request ref prior_request_auth_ref	*String* 36-character alphanumeric	Refers to the 3DS ACS Transaction ID in the response of the previous 3DS authentication.
prior request auth method prior_request_auth_ method	*String* 2-character numeric	Mechanism used by the cardholder to authenticate in the previous 3DS authentication: 01 = Frictionless authentication 02 = Challenge authentication 03 = AVS verified 04 = Other issuer methods
prior request auth timestamp prior_request_auth_ timestamp	*String* 12-character numeric	Date and time in UTC of the prior card-holder authentication. Found in the previous 3DS authentication response as 3DS Auth TimeStamp. Format is YYYYMMDDHHMM.
recurring expiry recurring_expiry	*String* 8-character numeric	End date after which no further recur-ring transactions shall be performed. Format is YYYYMMDD.

Variable Name	Type and Limits	Description
recurring frequency recurring_frequency	*String* 4-character numeric	The minimum number of days between recurring transactions. Numeric values between 1 and 9999, leading zeroes accepted.
request challenge request_challenge	*String* 2-character numeric	Indicates whether a browser-based challenge is requested for this transaction. Standard is "01" • 01 = No preference • 02 = No challenge requested • 03 = Challenge requested: 3DS Requestor Preference • 04 = Challenge requested: Mandate
request type request_type	*String* 2-character alphanumeric	Indicates the type of browser-based authentication request: 01 = cardholder initiated payment 02 = recurring transaction
shipping address ship_address1	*String* 50-character alphanumeric	Shipping destination address
ri indicator ri_indicator **NOTE:** Visa Secure only support ri_Indicator = 01, 02, 06, 07, or 11 for Payment Transactions and ri Indicator = 03, 04, 05 and 10 for Non Payment Transactions	*String* 2-character numeric	The type of 3DS Requestor Initiated (3RI) request: 01 = Recurring 02 = Installment 03 = Add Card 04 = Maintain Card Information 05 = Account verification 06 = Split/Delayed Shipment

Variable Name	Type and Limits	Description
		07 = Top-up 08 = Mail Order 09 = Telephone Order 10 = Whitelist 11 = Other Payment
shipping city ship_city	*String* 50-character alphanumeric	Shipping destination city
shipping country ship_country	*String* 3-character alphanumeric	Shipping destination country Defined as 3-digit country code in ISO 3166-1
shipping postal code ship_postal_code	*String* 16-character alphanumeric	Shipping destination postal or ZIP code
shipping province ship_province	*String* 3-character alphanumeric	Shipping destination province Defined in country subdivision ISO 3166-2
3DS completion indicator three_ds_completion_ind	*String* 1-character alphanumeric	indicates whether 3ds method MpiCardLookup was successfully completed Allowable values: Y = Successfully completed N = Did not successfully complete U = Unavailable
browser IP Address <browser_ip_address>	*String* Allows '.' and ':' 45-character alphanumeric	IP address of the browser as returned by the HTTP headers to the 3DS Requestor. **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide. Lack of providing this field, might

Variable Name	Type and Limits	Description
		increase the risk of rejects.
cardholder work phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Work]]]>	*Object* N/A	Cardholder work phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects. **NOTE:** This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.
cardholder home phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Home]]]>	*Object* N/A	Cardholder home phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects. **NOTE:** This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.
cardholder mobile phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Mobile]]]>	*Object* N/A	Cardholder mobile phone number **NOTE:** This field is not mandatory, but it is required. It is highly recommended to provide at least one of the Cardholder Phone Number. Lack of providing at least one of the Cardholder Phone Number, might increase the risk of rejects.

Variable Name	Type and Limits	Description
		NOTE: This is a nested object within the transaction. For information about fields in the Cardholder Phone Number Info object, see Cardholder Phone Number Info Object and Variables.

MPI 3DS Cardholder Phone Number

Variable Name	Type and Limits	Description
country code <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-CountryCode]]]>	*String* 3-character numeric	Country Code of phone number provided by the Cardholder.
phone number <[[[Undefined variable Global-Request-Field-s/Global-3DS-Req-Fld-s.Cardholder-Phone-Subscriber]]]>	*String* 15-character numeric	The phone number provided by the Cardholder.

A.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>	*Object* N/A	Contains fields applying to the Address Verification Service (AVS) e-fraud tool
Credential on File Information	*Object* N/A	Required when storing cardholder credentials or using these credentials in

Variable Name	Type and Limits	Description
<cof_info>		subsequent transactions.
Convenience Fee Information <convfee_info>	*Object* N/A	Contains fields related to the Convenience Fee feature
Customer Information <cust_info>	*Object* N/A	Contains fields that describe miscellaneous customer information, billing and shipping information, and item information
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

A.5 Definition of Request Fields – Credential on File

Variable Name	Type and Limits	Description
issuer ID <issuer_id>	*String* 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 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)

Variable Name	Type and Limits	Description
		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 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
payment indicator <payment_indicator>	*String* 1-character alphabetic	Indicates the current or intended use of the credentials Possible values for first transactions: C - unscheduled Credential on File (first transactions only) R - recurring V - recurring variable payment transaction Possible values for subsequent transactions: R - recurring V - recurring variable payment transaction U - unscheduled merchant-initiated transaction Z - unscheduled customer-initiated transaction 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: if payment indicator = R, then allowable values for e-commerce indicator: 2, 5 or 6 if payment indicator = V, 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
payment information <payment_information>	*String* 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)

A.6 Definition of Request Fields – Installments by Visa

Variable Name	Type and Limits	Description
Installment Info	*Object* N/A	Contains request fields related to installments
installment plan ID	*String* 36-character alphanumeric fixed length	Card brand-generated identifier for an installment plan
installment plan reference	*String* 10-character alphanumeric fixed length	Unique, human friendly name for the installment plan
terms and conditions version	*String* 10-character alphanumeric variable length (1-10 characters)	Version of the terms and conditions of the installment plan accepted by the cardholder The version is auto-incremented every time an update is made to the plan by the issuer

A.7 Definition of Request Fields – Apple Pay Token

Apple Pay Token transaction request fields – Required

Variable Name	Type and Limits	Description
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: • 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: Public Key Hash, Ephemeral Public Key, Transaction ID
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	*String*	Ephemeral public key bytes

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

Apple Pay Token transaction request fields – Optional

Variable Name	Type and Limits	Description
network <network>	*String* N/A	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 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	Description of the payment method type, contains the string representation of the PKPayment.paymentMethod.type. This field is required for ApplePay INTERAC transactions 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:

Variable Name	Type and Limits	Description
		3DSecure = Cryptogram obtained using MerchantCapability3DS EMV = Cryptogram obtained using MerchantCapablitiyEMV
token originator `<token_originator>`	*Object* N/A	This is used for merchants who owns multiple merchant accounts and would like to decrypt using the encryption key of a master store

A.8 Definition of Request Fields – GooglePay Token Temp Add

Variable Name	Type and Limits	Description
payment token `<payment_token>`	*Object* *N/A*	Payment details returned by Google in their `PaymentData` object for GooglePay transactions. See Definition of Request Fields – GooglePay Token Temp Add below for field details.
signature `<signature>`	*String*	Verifies that the message came from Google. It's base64-encoded, and created with ECDSA by the intermediate signing key. Returned by Google in their `PaymentData` object for GooglePay transactions
protocol version `<protocol_version>`	*String*	Identifies the encryption or signing scheme under which the message is created. It allows the protocol to evolve over time, if needed. Returned by Google in their `PaymentData` object for GooglePay transactions
signed message `<signed_message>`	*String*	A JSON object serialized as an HTML-safe string that contains the encryptedMessage, ephemeralPublicKey, and tag. It's serialized to simplify the signature verification process. Returned by Google in their `PaymentData` object for GooglePay transactions

A.9 Definition of Request Fields – Recurring Billing

Recurring Billing Info Object Request Fields

Variable Name	Type and Limits	Description
number of recurs <num_recurs>	*String* numeric 1-999	The number of times that the transaction must recur
period <period>	*String* numeric 1-999	Number of recur units 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 **EXAMPLE:** 1234567.89	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
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 imme-

Variable Name	Type and Limits	Description
		diately; 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

A.10 Definition of Request Fields – Account Name Verification Object

Request fields within the Account Name Verification object. The object can only be included in Card Verification transactions. Account name verification is only applicable to Visa credit cards.

Variable Name	Type and Limits	Description
First Name <first_name>	*String* 32-character alphanumeric	Cardholder last name
Middle Name <middle_name>	*String* 32-character alphanumeric	Cardholder middle name
Last Name <last_name>	*String* 32-character alphanumeric	Cardholder last name

A.11 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
AVS street name	*String*	Cardholder's address street name

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

A.12 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>	*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.

Appendix B Definitions of Response Fields

Table 4: Receipt object response values

Value	Type	Limits	Get Method
	Description		
General response fields			
Card type	String	2-character alphabetic (min. 1)	
	Represents the type of card in the transaction, e.g., Visa, Mastercard. Possible values: • V = Visa • M = Mastercard • AX = American Express • DC = Diner's Card • NO = Novus/Discover • SE = Sears • D = Debit • C1 = JCB		
Card level result	String	3-alphanumeric	
	(US only) Returns the product ID for the Visa or MC card program from the issuer. For a list of all Visa and MasterCard Card Level Result values refer to the Moneris developer portal at https://developer.moneris.com.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Transaction amount	String	10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point **EXAMPLE:** 1234567.89	
	Transaction amount that was processed.		
Transaction number	String	255-character alphanumeric	
	Gateway Transaction identifier often needed for follow-on transactions (such as Refund and Purchase Correction) to reference the originally processed transaction.		
Receipt ID	String	50-character alphanumeric	
	Order ID that was specified in the transaction request.		
Transaction type	String	2-character alphanumeric	
	• 0 = Purchase • 1 = Pre-Authorization • 2 = Completion • 4 = Refund • 11 = Void		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Reference number	String	18-character numeric	
	Terminal used to process the transaction as well as the shift, batch and sequence number. This data is typically used to reference transactions on the host systems, and must be displayed on any receipt presented to the customer. This information is to be stored by the merchant. Example: 660123450010690030 • 66012345: Terminal ID • 001: Shift number • 069: Batch number • 003: Transaction number within the batch.		
Response code	String	3-character numeric	
	• < 50: Transaction approved • ≥ 50: Transaction declined • Null: Transaction incomplete. For further details on the response codes that are returned, see the Response Codes document at https://developer.moneris.com.		
ISO	String	2-character numeric	
	ISO response code		
Bank totals	Object		
	Response data returned in a Batch Close and Open Totals request. See "Definitions of Response Fields" on page 352.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Message	String	100-character alpha-numeric	
	Response description returned from issuer. The message returned from the issuer is intended for merchant information only, and is **not** intended for customer receipts.		
Authorization code	String	8-character alphanumeric	
	Authorization code returned from the issuing institution.		
Complete	String	true/false	
	Transaction was sent to authorization host and a response was received		
Transaction date	String	Format: yyyy-mm-dd	
	Processing host date stamp		
Transaction time	String	Format: ##:##:##	
	Processing host time stamp		
Ticket	String	N/A	
	Reserved field.		
Timed out	String	true/false	
	Transaction failed due to a process timing out.		
Is Visa Debit	String	true/false	
	Indicates whether the card processed is a Visa Debit.		
PBBLifeCycleTraceID	String	15-alphanumeric	
	Unique transaction identifier from Interac Direct systems. Applies to Interac Direct transactions only and is used to link follow-on transactions.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Account Name Verification Result <AccountNameVerificationResult>	String	10-character alpha-numeric	
	Code indicating the results of Visa Account Name Verification. Position 1 and 2: Overall inquiry status. Position 3 and 4: Full name match status Position 5 and 6: last name match status Position 7 and 8: middle name match status Position 9 to 10: first name match status Inquiry status values: 00 = Name match performed 01 = Name match no performed 02 = Name match not supported Values for full name match and last/middle/first name match: 01 = Match 50 = Partial Match 99 = No Match		
Batch Close/Open Totals response fields			
Processed card types	String Array	N/A	
	Returns all of the processed card types in the current batch for the terminal ID/ECR Number from the request.		
Terminal IDs	String	8-character alpha-numeric	
	Returns the terminal ID/ECR Number from the request.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Purchase count	String	4-character numeric	
	Indicates the # of Purchase, ACH debit, Pre-Authorization Completion and Force Post transactions processed. If none were processed in the batch, then the value returned will be 0000.		
Purchase amount	String	11-character alpha-numeric	
	Indicates the dollar amount processed for Purchase, Pre-Authorization Completion or Force Post transactions. This field begins with a + and is followed by 10 numbers, the first 8 indicate the amount and the last 2 indicate the penny value. **EXAMPLE:** +0000000000 = 0.00 and +0000041625 = 416.25		
Refund count	String	4-character numeric	
	Indicates the # of Refund or Independent Refund transactions processed. If none were processed in the batch, then the value returned will be 0000.		
Refund amount	String	11-character alpha-numeric	
	Indicates the dollar amount processed for Refund, Independent Refund or ACH Credit transactions. This field begins with a + and is followed by 10 numbers, the first 8 indicate the amount and the last 2 indicate the penny value. Example, +0000000000 = 0.00 and +0000041625 = 416.25		
Correction count	String	4-character numeric	
	Indicates the # of Purchase Correction or ACH Reversal transactions processed. If none were processed in the batch, then the value returned will be 0000.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Correction amount	String	11-character alpha-numeric	
	Indicates the dollar amount processed for Purchase Correction transactions. This field begins with a + and is followed by 10 numbers, the first 8 indicate the amount and the last 2 indicate the penny value. **EXAMPLE:** +0000000000 = 0.00 and +0000041625 = 416.25		
Recurring Billing Response Fields (see Appendix A, page 1)			
Recurring billing success	String	true/false	
	Indicates whether the recurring billing transaction has been successfully set up for future billing.		
Recur update success	String	true/false	
	Indicates recur update success.		
Next recur date	String	yyyy-mm-dd	
	Indicates next recur billing date.		
Recur end date	String	yyyy-mm-dd	
	Indicates final recur billing date.		
Status Check response fields (see)			
Status code	String	3-character alpha-numeric	
	• < 50: Transaction found and successful • ≥ 50: Transaction not found and not successful **NOTE:** the status code is only populated if the connection object's Status Check property is set to **true**.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Status message	String	found/not found	
	• Found: 0 ≤ Status Code ≤ 49 • Not Found or null: 50 ≤ Status Code ≤ 999. **NOTE:** The status message is only populated if the connection object's Status Check property is set to **true**.		
AVS response fields (see Appendix A, page 1)			
AVS result code	String	1-character alpha-numeric	
	Indicates the address verification result. For a full list of possible response codes refer to Section Appendix B.		
CVD response fields (see)			
CVD result code	String	2-character alpha-numeric	
	Indicates the CVD validation result. The first byte is the numeric CVD indicator sent in the request; the second byte is the response code. Possible response codes are shown in Appendix B		
GooglePay Token response fields			
GooglePay Payment Method	String	4-character alpha-numeric	
	Indicates if the underlying card used in the GooglePay digital wallet is the funding card number ("FPAN") or a tokenized card number ("DPAN"). If a GoogleTokenTem-pAdd returns an FPAN, you may peform 3DS authen-tication with it; if it returns a DPAN, 3DS is not required.		
MPI response fields (see "MPI" on page 1)			
Type	String	99-character alpha-numeric	
	VERes, PARes or error defines what type of response you are receiving .		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Success	Boolean	true/false	
	True if attempt was successful, false if attempt was unsuccessful.		
Message	String	100-character alphabetic	
	MPI TXN transactions can produce the following values: • Y: Create VBV verification form popup window. • N: Send purchase or preauth with crypt type 6 • U: Send purchase or preauth with crypt type 7. MPI ACS transactions can produce the following values: • Y or A: (Also `receipt.getMpiSuccess()=true`) Proceed with cavv purchase or cavv preauth. • N: Authentication failed or high-risk transaction. It is recommended that you do not to proceed with the transaction. Depending on a merchant's risk tolerance and results from other methods of fraud detection, transaction may proceed with crypt type 7. • U or time out: Send purchase or preauth as crypt type 7.		
Term URL	String	255-character alphanumeric	
	URL to which the PARes is returned		
MD	String	1024-character alphanumeric	
	Merchant-defined data that was echoed back		
ACS URL	String	255-character alphanumeric	
	URL that will be for the generated pop-up		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
MPI CAVV	String	28-character alpha-numeric	
	VbV/MCSC/American Express SafeKey authentication data		
MPI E-Commerce Indicator	String	1-character alpha-numeric	
CAVV result code	String	1-character alpha-numeric	
	Indicates the Visa CAVV result. For more information, see 1 Cavv Result Codes for Verified by Visa. • 0 = CAVV authentication results invalid • 1 = CAVV failed validation; authentication • 2 = CAVV passed validation; authentication • 3 = CAVV passed validation; attempt • 4 = CAVV failed validation; attempt • 7 = CAVV failed validation; attempt (US issued cards only) • 8 = CAVV passed validation; attempt (US issued cards only) • The CAVV result code indicates the result of the CAVV validation.		
MPI inline form			
Vault response fields (see 4.1, page 73)			

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Data key	String	28-character alpha-numeric	
	The data key response field is populated when you send a Vault Add Credit Card- ResAddCC (page 1), Vault Encrypted Add Credit Card - EncResAddCC (page 1), Vault Tokenize Credit Card - ResTokenizeCC (page 1), Vault Add Temporary Token - ResTempAdd (page 1) or Vault Add Token - ResAddToken (page 1) transaction. It is the profile identifier that all future financial Vault transactions will use to associate with the saved information.		
Vault payment type	String	cc	
	Indicates the payment type associated with a Vault profile		
Expiring card's Payment type	String	cc	
	Indicates the payment type associated with a Vault profile.		
Vault masked PAN	String	20-character numeric	
	Returns the first 4 and/or last 4 of the card number saved in the profile.		
Expiring card's Masked PAN	String	20-character numeric	
	Returns the first 4 and/or last 4 of the card number saved in the profile.		
Vault success	String	true/false	
	Indicates whether Vault transaction was successful.		
Vault customer ID	String	30-character alpha-numeric	
	Returns the customer ID saved in the profile.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Expiring card's customer ID	String	30-character alpha-numeric	
	Returns the customer ID saved in the profile.		
Vault phone number	String	30-character alpha-numeric	
	Returns the phone number saved in the profile.		
Expiring card's phone number	String	30-character alpha-numeric	
	Returns the phone number saved in the profile.		
Vault email address	String	30-character alpha-numeric	
	Returns the email address saved in the profile.		
Expiring card's email address	String	30-character alpha-numeric	
	Returns the email address saved in the profile.		
Vault note	String	30-character alpha-numeric	
	Returns the note saved in the profile.		
Expiring card's note	String	30-character alpha-numeric	
	Returns the note saved in the profile.		
Vault expiry date	String	4-character numeric	
	Returns the expiry date of the card number saved in the profile. YYMM format.		
Expiring card's expiry date	String	4-character numeric	
	Returns the expiry date of the card number saved in the profile. YYMM format.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Vault E-commerce indicator	String	1-character numeric	
	Returns the e-commerce indicator saved in the profile.		
Expiring card's E-commerce indicator	String	1-character numeric	
	Returns the e-commerce indicator saved in the profile.		
Vault AVS street number	String	19-character alpha-numeric	
	Returns the AVS street number saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		
Expiring card's AVS street number	String	19-character alpha-numeric	
	Returns the AVS street number saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		
Vault AVS street name	String	19-character alpha-numeric	
	Returns the AVS street name saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		
Expiring card's AVS street name	String	19-character alpha-numeric	
	Returns the AVS street name saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		
Vault AVS ZIP code	String	9-character alpha-numeric	
	Returns the AVS zip/postal code saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Expiring card's AVS ZIP code	String	9-character alpha-numeric	
	Returns the AVS zip/postal code saved in the profile. If no other AVS street number is passed in the transaction request, this value will be submitted along with the financial transaction to the issuer.		
Vault customer first name	String	50-character alpha-numeric	
	(US ACH only) Returns the customer first name saved in the profile.		
Vault customer last name	String	50-character alpha-numeric	
	(US ACH only) Returns the customer last name saved in the profile.		
Vault customer address 1	String	50-character alpha-numeric	
	(US ACH only) Returns the customer address line 1 saved in the profile.		
Vault customer address 2	String	50-character alpha-numeric	
	(US ACH only) Returns the customer address line 2 saved in the profile.		
Vault customer city	String	50-character alpha-numeric	
	US ACH only Returns the customer city saved in the profile.		
Vault customer state	String	2-character alpha-numeric	
	US ACH only Returns the customer state code saved in the profile.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Vault customer ZIP code	String	10-character numeric	
	US ACH only Returns the customer zip code saved in the profile.		
Vault check routing number	String	9-character numeric	
	US ACH only Returns the customer check routing number saved in the profile.		
Vault masked account number	String	15-character alpha-numeric	
	US ACH only Returns the masked first 4 and last 4 digits of the account number saved in the profile.		
Vault account number	String	15-character alpha-numeric	
	US ACH only Returns the full account number saved in the profile. Applicable to Vault Lookup Full transaction type only.		
Vault check number	String	16-character numeric	
	US ACH only Returns the check number saved in the profile.		
Vault account type	String	savings/checking	
	US ACH only Returns the type of account saved in the profile.		
Vault SEC code	String	3-character alpha-numeric	
	US ACH only Returns the ACH SEC code saved in the profile.		
Vault credit card number	String	20-character numeric	
	Returns the full credit card number saved in the Vault profile. Applicable to Vault Lookup Full transaction only.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Corporate card	String	true/false	
	Indicates whether the card associated with the Vault profile is a corporate card.		
Encrypted Mag Swipe response fields (see Section 1, page 1)			
Masked credit card number	String	20-character alpha-numeric	
Convenience Fee response fields (see Appendix A, page 1)			
Convenience fee success	String	true/false	
	Indicates whether the Convenience Fee transaction processed successfully.		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Convenience fee status	String	3-character alpha-numeric	
	Indicates the status of the merchant and convenience fee transactions. The CfStatus field provides details about the transaction behavior and should be referenced when contacting Moneris Customer Support. Possible values are: • 1 or 1F – Completed 1st purchase transaction • 2 or 2F – Completed 2nd purchase transaction • 3 – Completed void transaction • 4A or 4D – Completed refund transaction • 7 or 7F – Completed merchant independent refund transaction • 8 or 8F – Completed merchant refund transaction • 9 or 9F – Completed 1st void transaction • 10 or 10F – Completed 2nd void transaction • 11A or 11D – Completed refund transaction		
Convenience fee amount	String	9-character decimal	
	The expected Convenience Fee amount. This field will return the amount submitted by the merchant for a successful transaction. For an unsuccessful transaction, it will return the expected convenience fee amount		

Table 4: Receipt object response values (continued)

Value	Type	Limits	Get Method
	Description		
Convenience fee rate	String	9-character decimal	
	The convenience fee rate that has been defined on the merchant's profile. For example: 1.00 – a fixed amount or 10.0 - a percentage amount		
Convenience fee type	String	AMT/PCT	
	The type of convenience fee that has been defined on the merchant's profile. Available options are: AMT – fixed amount PCT – percentage		
Merchant Advice Code response field			
Advice Code	String	2-character alpha-numeric	
	The message returned from the issuer is intended for merchant information only, and is not intended for customer receipts. For further details on the response codes that are returned, see the Advice Code document at https://developer.moneris.com.		

Table 5: Financial transaction response codes

Code	Description
< 50	Transaction approved
≥ 50	Transaction declined
NULL	Transaction was not sent for authorization

For more details on the response codes that are returned, see the Response Codes document available at https://developer.moneris.com

Table 6: Vault Admin Responses

Code	Description
001	Successfully registered CC details. Successfully updated CC details. Successfully deleted CC details. Successfully located CC details. Successfully located # expiring cards. (NOTE: # = the number of cards located)
983	Cannot find previous
986	Incomplete: timed out
987	Invalid transaction
988	Cannot find expiring cards
Null	Error: Malformed XML

B.1 Definition of Response Fields –Installments by Visa

Response fields appearing in the Installment Plan Lookup transaction

Variable Name	Type and Limits	Description
Eligible Installment Plans	*Object* N/A	Contains fields related to the installment plan
plan count	*String* numeric	Total number of installment plans available for offer to the cardholder
Plan Details	*Array object* N/A	Contains fields related to the particular installment plan Each installment plan on offer to the

Variable Name	Type and Limits	Description
		cardholder is represented by a distinct Plan Details object
annual percentage rate (APR)	*String* numeric	Annual percentage rate (APR) attached to the installment plan payments; for display purposes only and not used for calculations Allowable values: 0-10000 Percentage rate is represented with two implicit decimals **EXAMPLE:** 320 is 3.2%
installment frequency	*String* max 10-character alphabetic	Frequency of installments for the plan Potential values: WEEKLY BIWEEKLY MONTHLY BIMONTHLY
installment plan ID	*String* 36-character alphanumeric fixed length	Card brand-generated identifier for an installment plan Used as a request field in the Installment Info object
installment plan name	*String* max 255-character alphanumeric	Name of the installment plan; may not be unique
installment plan reference	*String* 10-character alphanumeric fixed length	Unique, human friendly name for the installment plan Used as a request field in the Installment Info object
installment plan type	*String* max 20 character alpha-	Type of installment plans Potential values:

Variable Name	Type and Limits	Description
	numeric	ISSUER_PROMOTION BI_LATERAL ISSUER_DEFAULT MARKET
number of installments	*String* 4-character numeric min 1, max 1000	Maximum number of installments in the plan
First Installment	*Object* N/A	Contains cost details for the first installment
first installment amount	*String* max 9-character numeric	Amount of the first installment payment Final two digits represent penny values **EXAMPLE:** 123112 = $1231.12
first installment fee	*String* max 9-character numeric	Fee charged on the first installment Final two digits represent penny values **EXAMPLE:** 123112 = $1231.12
upfront fee	*String* numeric	The up-front fee charged to the cardholder for the installment plan; only charged on the first installment
Last Installment	*Object* N/A	Contains cost details for the last installment
last installment amount	*String* max 9-character numeric	Amount of the final installment payment

Variable Name	Type and Limits	Description
		Final two digits represent penny values **EXAMPLE:** 123112 = $1231.12
last installment fee	*String* max 9-character numeric	Fee charged on the last installment Final two digits represent penny values **EXAMPLE:** 123112 = $1231.12
Promotion Info	*Object* N/A	Contains promotion information shared between the issuer and the merchant
promotion code	*String* 2-character alphanumeric	An external identifier for the plan provided by the issuer
promotion ID	*String* max 8-character alphanumeric	An external identifier provided by the issuer that identifies a program or promotion
Terms and Conditions	*Array object* N/A	Contains fields related to terms and conditions presented to the cardholder
terms and conditions count	*String* numeric	Number of instances of the set of terms and conditions attached to a particular installment plan, representing the number of languages they are offered in
Terms and Conditions Details	*Object* N/A	Contains details related to a particular language set (English, French, etc.) of terms and conditions being offered Each language set has its own object
language code	*String*	Language code for the terms and conditions text

Variable Name	Type and Limits	Description
	3-character alphanumeric	
text	*String* max 2000-character alpha-numeric	Text of the terms and conditions for the installment plan
terms and conditions URL	*String* max 1000 character-alpha-numeric	A terms and conditions HTTPS URL hosted by the issuer for displaying to the cardholder
terms and conditions version	*String* 10-character alphanumeric variable length (1-10 characters)	Version of the terms and conditions of the installment plan accepted by the cardholder The version is auto-incremented every time an update is made to the plan by the issuer
total fees	*String* max 9-character numeric	Total fees charged by the plan Final two digits represent penny values **EXAMPLE:** 123112 = $1231.12
total plan cost	*String* numeric	Represents the total amount the selected installment plan will cost The right-most digits represent minor units (e.g., cents in CAD); no fractional minor units **EXAMPLE:** 123112 in CAD represents CAD $1231.12

Response fields appearing in financial transactions

Variable Name	Type and Limits	Description
Installment Results	*Object*	Contains fields related to the install-

Variable Name	Type and Limits	Description
	N/A	ment plan in financial transactions
installment plan ID	*String* 36-character alphanumeric fixed length	Card brand-generated identifier for an installment plan
installment plan reference	*String* 10-character alphanumeric fixed length	Unique, human friendly name for the installment plan
terms and conditions version	*String* 10-character alphanumeric variable length (1-10 characters)	Version of the terms and conditions of the installment plan accepted by the cardholder. The version is auto-incremented every time an update is made to the plan by the issuer.
plan acceptance ID	*String* 36-character alphanumeric fixed length	Visa-generated, alphanumeric, unique and short human-readable name for the installment plan
installment plan status	*String* 1-character alphabetic fixed length	Potential values: N – new plan, not accepted yet A – accepted plan C – cancelled plan
plan response	*String* max 50-character numeric	Response code for the installment plan Potential values: 00 – processed and approved If not 00, indicates installment plan processing failure; a verbose error response message as received from from Visa is returned