



XML Requester Service

Friday, June 19th, 2020

For support, please contact:
(800) 929-1848
<http://www.quantumgateway.com>

Table of Contents

Request URL – XML	4
Request URL – HTTPS POST	4
Request Wrapper	4
QGWRequest	4
Authentication.....	5
Vault vs. Virtual Terminal	5
Request	5
Response Wrapper	5
QGWRequest	5
ResponseSummary	6
Result	6
Request and Response Examples	7
Sample Request	7
Sample Response	7
Sample Error Response	7
Request Listing - General	8
Authentication.....	8
ResponseSummary	8
Request Listing – Transactions	9
SearchTransactions.....	9
ShowTransactionDetails	11
Create Vault Debit/Credit Transaction	14
Process Single Credit/Debit Transactions	16
Partial Authorizations	20
What is Partial Authorization?	20
Void Partial Auth	21
Force Partial Auth.....	22
Resubmit Transaction	23
ResubmitTransaction	23
Request Listing – Vault	25
SearchVault	25
ShowVaultDetails	26
Request Listing – Customers	27
CustomerAdd	27
CustomerUpdate	29
CustomerUpdatePayOnly	31
* Updates the credit card number and expiration date or updates an EFT routing and account number.	31
AddUpdateCustomer	32
CustomerRemove	34
ExpiringVaultCustomers	35
ExpiringRecurringCustomers	36
SearchRecurring	37
ShowRecurringCustomer	38
UpdateRecurringCustomer	39
Vault To Recurring	40
Create Recurring Transactions From Vault	40
Maxmind Definitions	42
Sample PeopleCode	43
Sample Send Synchronous Message	43
Sample PHP Code	44
Sample CURL POST	44
Sample Xml Parsing.....	44
Request Errors	46
All RequestTypes	46
AddUpdateCustomer	46
CreateTransaction	46

CustomerAdd	46
CustomerRemove	46
CustomerUpdate	46
ExpiringVaultCustomers	47
ProcessSingleTransaction	47
ResubmitTransaction	48
SearchRecurring.....	48
SearchTransactions.....	48
SearchVault	48
ShowRecurringCustomer	48
ShowVaultDetails	48
UpdateRecurringCustomer	48
VaultCheck	48
VaultCreateRecurring	48
Field Index	48

Request URL – XML

Live System:

All Requester transactions must be submitted to the following URL:

https://secure.quantumgateway.com/cgi/xml_requester.php

GatewayLogin: Your Quantum Gateway Gateway ID

GatewayKey: Your Quantum Gateway Vault Key or Restrict Key

NOTE: For testing supply your test GatewayLogin and GatewayKey

Request URL – HTTPS POST

Live System:

All Requester transactions must be submitted to the following URL:

https://secure.quantumgateway.com/cgi/xml_requester.php

GatewayLogin: Your Quantum Gateway Gateway ID

GatewayKey: Your Quantum Gateway Vault Key or Restrict Key

Request Parameter: ?xml=

NOTE: For testing supply your test GatewayLogin and GatewayKey

Request Wrapper

QGWRequest

The outer tag for the Requester XML request. This tag must be the outermost XML element.

```
<QGWRequest>
  ..
  request details here
  ..
</QGWRequest>
```

Which Key To use:

VaultKey

AddCustomer, CreateTransaction, CustomerRemove, CustomerUpdate, ExpiringVaultCustomers, SearchVault, ShowVaultDetails, VaultCheck, VaultCreateRecurring, AddUpdateRecurring

RestrictKey

ProcessSingleTransaction, RecurringHistory, SearchRecurring, SearchTransactions, ShowRecurringCustomer, ShowTransactionDetails, UpdateRecurringCustomer

VaultKey is Located, and can be set, under "Edit Vault Config" in the Secure Vault of our gateway.
RestrictKey is located and can be set under RestrictKey in the Processing Settings of our gateway.
GatewayLogin is provided to you at the time your account was created.

Authentication

The Authentication tag for requests is a child of the QGWRequest tag. It contains two child elements. GatewayLogin and GatewayKey. Every request must have an authentication tag with valid contents in its two child elements.

Vault vs. Virtual Terminal

Parameter Name	Type	Required	Vault	Virtual Terminal
GatewayLogin	(string)	Yes	Gateway ID	Gateway ID
GatewayKey	(string)	Yes	Vault Key	Restrict Key

```
<QGWRequest>
  <Authentication>
    <GatewayLogin>mygatewaylogin</GatewayLogin>
    <GatewayKey>mygatewaykey</GatewayKey>
  </Authentication>
  ..
  details here
  ..
</QGWRequest>
```

Request

The Request tag must be included in all requests. The RequestType selects the command for the Requester. For details of each Request along with the parameters of the RequestType, see the Request Listings.

```
<QGWRequest>
  ..
  authentication
  ..
  <Request>
    <RequestType>ShowTransactionDetails</RequestType>
    <TransactionID>322282603</TransactionID>
  </Request>
</QGWRequest>
```

Response Wrapper

Each request in a transmission will return an associated response result in the response transmission. The response wrappers structure is similar to the request wrapper.

QGWRequest

The outer tag for the Requester XML response. This tag will be the outermost XML element.

```
<QGWRequest>
  ..
  response details here
  ..
</QGWRequest>
```

ResponseSummary

The responssummary tag is a child of QGWRequest. It contains several child elements pertaining to the results of the request.

```
<QGWRequest>
  <ResponseSummary>
    <RequestType>SearchTransactions</RequestType>
    <Status>Success</Status>
    <StatusDescription>Request was successful.</StatusDescription>
    <ResultCount>1</ResultCount>
    <TimeStamp>8/20/2003 8:17:10 AM</TimeStamp>
  </ResponseSummary>
  ..
  response details here
  ..
</QGWRequest>
```

Result

The result tag is a child of QGWRequest. It contains several child elements pertaining to the results of the request.

```
<QGWRequest>
  ..
  response summary details here
  ..
  <Result>
    <TransactionID>123456</TransactionID>
    <Status>SETTLED</Status>
    <ACHTransactionID/>
    <ACHStatus/>
    <TransactionDate>2025-02-02</TransactionDate>
  </Result>
  <Result>
    <TransactionID>123456</TransactionID>
    <Status>SETTLED</Status>
    <ACHTransactionID/>
    <ACHStatus/>
    <TransactionDate>2025-02-02</TransactionDate>
  </Result>
  ..
  zero or more results here
  ..
</QGWRequest>
```

Request and Response Examples

Sample Request

```
<QGWRequest>
  <Authentication>
    <GatewayLogin>mygatewaylogin</GatewayLogin>
    <GatewayKey>mygatewaykey</GatewayKey>
  </Authentication>
  <Request>
    <RequestType>ShowTransactionDetail</RequestType>
    <TransactionID>2282603</TransactionID>
  </Request>
</QGWRequest>
```

Sample Response

```
<QGWRequest>
  <ResponseSummary>
    <RequestType>ShowTransactionDetails</RequestType>
    <Status>Success</Status>
    <StatusDescription>Request was successful.</StatusDescription>
    <ResultCount>1</ResultCount>
    <TimeStamp>8/20/2003 8:34:28 AM</TimeStamp>
  </ResponseSummary>
  <Result>
    <TransactionID>123456789</TransactionID>
    <TransactionType>CREDIT</TransactionType>
    <TransactionDate>2025-02-05</TransactionDate>
    <Status>APPROVED</Status>
    <ACHTransactionID/>
    <ACHStatus>SETTLED</ACHStatus>
    <Amount>150.00</Amount>
    <CustomerID>414000000</CustomerID>
    <FirstName>John</FirstName>
    <LastName>Doe</LastName>
    <Address>1706 W. 182nd Street</Address>
    <ZipCode>23320</ZipCode>
    <EmailAddress>John@home.com</EmailAddress>
    <Memo>6812300000</Memo>
    <PaymentType>CC</PaymentType>
    <CardType>VI</CardType>
    <CreditCardNumber>5454</CreditCardNumber>
    <ExpireMonth>12</ExpireMonth>
    <ExpireYear>2020</ExpireYear>
    <ABANumber/>
    <AccountNumber/>
    <SettlementBatchID>780</SettlementBatchID>
    <SettlementDate>2025-02-05</SettlementDate>
  </Result>
</QGWRequest>
```

Sample Error Response

```
<QGWRequest>
  <ResponseSummary>
    <RequestType>ShowTransactionDetails</RequestType>
    <Status>Error</Status>
    <StatusDescription>Transaction not found.</StatusDescription>
    <ResultCount>0</ResultCount>
    <TimeStamp>8/20/2003 8:36:07 AM</TimeStamp>
  </ResponseSummary>
</QGWRequest>
```

```
</ResponseSummary>
</QGWRequest>
```

Request Listing - General

Authentication

Required in order to validate access to information.

Parameters

GatewayLogin (string)
GatewayKey (string)

Sample Request

```
<Authentication>
  <GatewayLogin>mygatewaylogin</GatewayLogin>
  <GatewayKey>mygatewaykey</GatewayKey>
</Authentication>
```

ResponseSummary

Used to display summary of request results.

ResponseSummary Parameters

TransactionType (string)
Status (string)
StatusDescription (string)
ResultCount (int)
TimeStamp (datetime)

Sample ResponseSummary

```
<ResponseSummary>
  <RequestType>ShowTransactionDetail</RequestType>
  <Status>Error</Status>
  <StatusDescription>Transaction not found.</StatusDescription>
  <ResultCount>0</ResultCount>
  <TimeStamp>8/20/2003 8:34:28 AM</TimeStamp>
</ResponseSummary>
```

NOTE: MasterCard now prohibits showing the expiration date for any of their cards. Any of the requests below that show expiration date will not be included in a response if that card is a MasterCard.

Request Listing – Transactions

SearchTransactions

Used as a search engine to find transactions based on request parameters.

If no Request Parameter is supplied all transactions are returned.

**** Not recommended if there are many transactions as it may timeout!**

Request Parameters	Required	Type	Notes
Status	No	(string)	PENDING/APPROVED/DECLINED
TransactionDateBegin	No	(datetime)	YYYY-MM-DD Format
TransactionDateEnd	No	(datetime)	YYYY-MM-DD Format
TransactionType	No	(string)	DEBIT/CREDIT for EFT CREDIT/RETURN for CC
CustomerID	No	(string)	
PaymentType	No	(string)	CC/EFT
SettlementBatchID	No	(string)	
SettlementDateBegin	No	(datetime)	YYYY-MM-DD Format
SettlementDateEnd	No	(datetime)	YYYY-MM-DD Format
RecurrID	No	(integer)	Recurring ID from Recurring Search
FirstName	No	(string)	
LastName	No	(string)	
Address	No	(string)	
City	No	(string)	
State	No	(string)	
ZipCode	No	(string)	
Company	No	(string)	
Country	No	(string)	
EmailAddress	No	(string)	
SFNAME	No	(string)	
SLNAME	No	(string)	
SADDR1	No	(string)	
SCITY	No	(string)	
SSTATE	No	(string)	
SZIP	No	(string)	
SCOUNTRY	No	(string)	
Amount	No	(currency)	
AuthorizationCode	No	(string)	
CreditCardNumber	No	(integer)	Last4 of Credit Card Number
RID	No	(integer)	Recurring Recipe ID

Sample Request

```
<Request>
  <RequestType>SearchTransactions</RequestType>
  <Status/>
  <TransactionDateBegin>2025-02-02</TransactionDateBegin>
  <TransactionDateEnd>2025-02-02</TransactionDateEnd>
  <TransactionType/>
  <CustomerID/>
  <PaymentType/>
  <SettlementBatchID/>
  <SettlementDateBegin/>
  <SettlementDateEnd/>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(string)	
Status	(string)	
ACHTransactionID	(string)	
ACHStatus	(string)	
ACHStatusDescription	(string)	
TransactionDate	(datetime)	YYYY-MM-DD Format
Amount	(currency)	
CustomerID	(string)	
TransactionType	(string)	
PaymentType	(string)	
ProcessType	(string)	
InvoiceNumber	(string)	

Sample Response

```
<Result>
  <TransactionID>9999991234</TransactionID>
  <TransactionDate>2020-01-01</TransactionDate>
  <TransactionTime>23:54:50</TransactionTime>
  <Status>APPROVED</Status>
  <ACHTransactionID> </ACHTransactionID>
  <ACHStatus></ACHStatus>
  <ACHStatusDescription>$LogDetails</ACHStatusDescription>
  <Amount>10.00 </Amount>
  <CustomerID>CD456</CustomerID>
  <SettlementBatchID>986512</SettlementBatchID>
  <SettlementDate>2020-01-02 01:00:00</SettlementDate>
  <TransactionType>AUTH_FORCE</TransactionType>
  <PaymentType>CC</PaymentType>
  <ProcessType>AUTH_CAPTURE</ProcessType>
  <InvoiceNumber>PO 123</InvoiceNumber>
</Result>
```

ShowTransactionDetails

Used to show all details of a specific transaction

Request Parameters	Required	Type	Notes
TransactionID	Yes	(string)	

Sample Request

```
<Request>
  <RequestType>ShowTransactionDetails</RequestType>
  <TransactionID>123456789</TransactionID>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
TransactionType	(string)	DEBIT/CREDIT
TransactionDate	(datetime)	
Status	(string)	APPROVED/DECLINED/ERROR
StatusDescription	(string)	
ACHTransactionID	(string)	
ACHStatus	(string)	PENDING/RETURNED/SETTLED/TRANSMITTED
ACHStatusDescription	(string)	Log details of status. ie reason for return
Amount	(currency)	
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
EmailAddress	(string)	
RecurrID	(integer)	Recurring ID of Customer
RID	(integer)	Recurring Recipe ID
Phone	(string)	
Company	(string)	
Memo	(string)	
PaymentType	(string)	CC/EFT
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 Digits of the credit card number. Only returned if PaymentType = CC
ExpireMonth	(integer)	Only returned if PaymentType = CC
ExpireYear	(integer)	Only returned if PaymentType = CC
ABANumber	(string)	Only returned if PaymentType = EFT
AccountNumber	(string)	Last 4 Digits of the bank account number. Only returned if PaymentType = EFT
SettlementBatchID	(integer)	
SettlementDate	(datetime)	
BatchedOut	(string)	Y/N/V/T
AuthorizationCode	(string)	
CVV2ResponseCode	(string)	
AVSResponseCode	(string)	
SFNAME	(string)	
SLNAME	(string)	
SADDRESS	(string)	
SCITY	(string)	
SSTATE	(string)	
SZIP1	(string)	
SCOUNTRY	(string)	

InvoiceDescription	(string)	
InvoiceNumber	(string)	
Maxmind Information – If Used		See MaxMind Definitions
Distance	(string)	
countryMatch	(string)	
countryCode	(string)	
freeMail	(string)	
anonymousProxy	(string)	
score	(string)	
proxyScore	(string)	
spamScore	(string)	
ip_region	(string)	
ip_city	(string)	
ip_latitude	(string)	
ip_longitude	(string)	
ip_isp	(string)	
ip_org	(string)	
max_billed	(string)	
max_date	(string)	
binMatch	(string)	
binCountry	(string)	
err	(string)	
binName	(string)	
binNameMatch	(string)	
binPhoneMatch	(string)	
binPhone	(string)	
custPhoneInBillingLoc	(string)	
highRiskCountry	(string)	
queriesRemaining	(string)	
cityPostalMatch	(string)	
shipCityPostalMatch	(string)	
isTransProxy	(string)	
carderEmail	(string)	
shipForward	(string)	
highRiskUsername	(string)	
highRiskPassword	(string)	
explanation	(string)	
riskScore	(string)	
max_decline_reason	(string)	

Sample Response

```
<Result>
  <TransactionID>123456789</TransactionID>
  <TransactionType>CREDIT</TransactionType>
  <TransactionDate>2025-02-15</TransactionDate>
  <Status>SETTLED</Status>
  <StatusDescription>Transaction is Complete.</StatusDescription>
  <ACHTransactionID/>
  <ACHStatus/>
  <ACHStatusDescription/>
  <Amount>150.00</Amount>
  <CustomerID>414000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <EmailAddress>John@home.com</EmailAddress>
  <Memo>6812300000</Memo>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <ABANumber/>
  <AccountNumber/>
  <SettlementBatchID>780</SettlementBatchID>
  <SettlementDate>2025-02-16 23:00:00</SettlementDate>
  <AuthorizationCode>55468</AuthorizationCode>
  <CVV2ResponseCode/>
  <AVSResponseCode/>
</Result>
```

Create Vault Debit/Credit Transaction

Used to create an individual Debit or Credit Transaction from Vault Customers

Request Parameters

Parameter Name	Required	Type	Notes
TransactionType	Yes	(string)	CREDIT for CC, DEBIT for EFT
CustomerID	Yes	(string)	Identifies the vault customer
Memo	No	(string)	Customer Defined field
Amount	Yes	(currency)	
TransactionDate	No	(datetime)	Defaults to transaction creation Date
ProcessType	No	(string)	AUTH_CAPTURE, AUTH_ONLY, SALES Default is SALES (EFT SALES only)

Sample Request

```
<Request>
  <RequestType>CreateTransaction</RequestType>
  <TransactionType>CREDIT</TransactionType>
  <CustomerID>40441000</CustomerID>
  <Memo>68123000</Memo>
  <Amount>19.95</Amount>
  <TransactionDate>2025-02-02</TransactionDate>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
TransactionType	(string)	DEBIT/CREDIT
TransactionDate	(datetime)	
Status	(string)	APPROVED/DECLINED/ERROR
StatusDescription	(string)	
ACHTransactionID	(string)	
ACHStatus	(string)	PENDING/RETURNED/SETTLED/TRANSMITTED
ACHStatusDescription	(string)	Log details of status. ie reason for return
Amount	(currency)	
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
EmailAddress	(string)	
Memo	(string)	
PaymentType	(string)	CC/EFT
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 Digits of the credit card number. Only returned if PaymentType = CC
ExpireMonth	(integer)	Only returned if PaymentType = CC
ExpireYear	(integer)	Only returned if PaymentType = CC
ABANumber	(string)	Only returned if PaymentType = EFT
AccountNumber	(string)	Last 4 Digits of the bank account number. Only returned if PaymentType = EFT
SettlementBatchID	(integer)	
SettlementDate	(datetime)	

AuthorizationCode	(string)	
CVV2ResponseCode	(string)	
AVSResponseCode	(string)	
RecurrID	(integer)	Unique Recurring ID
EnablePartialAuths	(integer)	Yes/No

Sample Response

```

<Result>
  <TransactionID>123456789</TransactionID>
  <TransactionType>CREDIT</TransactionType>
  <TransactionDate>2025-02-15</TransactionDate>
  <Status>SETTLED</Status>
  <StatusDescription>Transaction is Complete.</StatusDescription>
  <ACHTransactionID/>
  <ACHStatus/>
  <ACHStatusDescription/>
  <Amount>150.00</Amount>
  <CustomerID>414000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <EmailAddress>John@home.com</EmailAddress>
  <Memo>6812300000</Memo>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <ABANumber/>
  <AccountNumber/>
  <SettlementBatchID>780</SettlementBatchID>
  <SettlementDate>2025-02-16 23:00:00</SettlementDate>
  <AuthorizationCode>55468</AuthorizationCode>
  <CVV2ResponseCode/>
  <AVSResponseCode/>
  <RecurrID>55</RecurrID>
</Result>

```

Process Single Credit/Debit Transactions

Used to create an individual Debit/Credit Transaction.

Request Parameters

Parameter Name	Required	Type	Notes
TransactionID	(integer)		
TransactionType	Yes	(string)	DEBIT/CREDIT
ProcessType	No	(string)	RETURN
			AUTH_CAPTURE – Auth Then Sales
			AUTH_ONLY – Auth only
			SALES (default) – Skips AVS and CVV2
			*VOID – Requires TransactionID
			*PREVIOUS_SALE – Force An AUTH
PaymentType	Yes	(string)	CC/EFT
Amount	Yes	(currency)	Payment amount
CreditCardNumber	No	(string)	Only required for PaymentType CC
ExpireMonth	No	(string)	Only required for PaymentType CC
ExpireYear	No	(string)	Only required for PaymentType CC
CVV2	No	(string)	
ABANumber	No	(string)	Only required for PaymentType EFT
AccountNumber	No	(string)	Only required for PaymentType EFT Customer's checking or savings account Number
EFTType	No	(string)	Only required for PaymentType EFT **
AccountType	No	(string)	Checking/Savings
Memo	No	(string)	User defined field – For Internal Notes
FirstName	No	(string)	Customer's first name as it is on the Credit
LastName	No	(string)	Customer's last name as it is on the Credit
Address	No	(string)	Customer's billing address
City	No	(string)	Customer's billing city
State	No	(string)	Customer's billing state
ZipCode	No	(string)	Customer's billing zipcode
Country	No	(string)	Customer's billing Country
EmailAddress	No	(string)	Customer's email address
PhoneNumber	No	(string)	Customer's phone number
IPAddress	No	(string)	Customer's IP address
InvoiceNumber	No	(string)	Invoice number
InvoiceDescription	No	(string)	Invoice description
TransactionDate	No	(datetime)	Defaults to transaction creation Date
RID	No	(integer)	Recipe ID
CustomerID	No	(string)	Customer ID
RecurAmount	No	(currency)	Override Recuring Amount
RecurTimes	No	(integer)	Number of times to recur, 0 for no limit
MerchantEmail	No	(integer)	Y/N (Y will always send email)
CustomerEmail	No	(integer)	Y/N (Y will always send email)
OverRideRecurDay	No	(string)	Y/N (Y will set the recurring day to current date)
RequestType2	No	(string)	***CustomerAdd/CustomerUpdate
EnablePartialAuths	No	(integer)	Yes/No
PartialPaymentID	No	(integer)	id assigned on original transaction

* Requires TransactionID

** Required if your account was approved for multiple Sec Codes. Options are WEB,CCD,PPC and TEL.

*** Adds or updates customer to the vault after an approved transaction. CustomerID is required.

Response contains <Status2></Status2>

Status2 responses: No CustomerID Was Submitted, ALREADY In Vault, Added To Vault

Sample Request

```
<Request>
  <RequestType>ProcessSingleTransaction</RequestType>
  <TransactionType>CREDIT</TransactionType>
  <PaymentType>CC</PaymentType>
  <Amount>19.95</Amount>
  <CreditCardNumber>5454545454545454</CreditCardNumber>
  <PaymentType>CC</PaymentType>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <CVV2>456</CVV2>
  <ABANumber/>
  <AccountNumber/>
  <AccountType/>
  <Memo>6812300000</Memo>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <City>Aberdeen</City>
  <State>WA</State>
  <ZipCode>23320</ZipCode>
  <Country>USA</Country>
  <EmailAddress>John@home.com</EmailAddress>
  <PhoneNumber>3605381916</PhoneNumber>
  <IPAddress>216.223.17.34</IPAddress>
  <InvoiceNumber/>
  <InvoiceDescription/>
  <TransactionDate>2025-02-02</TransactionDate>
</Request>
```

Sample Request 2 – Return on a previously settled transaction

This sample shows how you can do a RETURN by only using the last 4 of the CreditCardNumber and the transaction ID. You can also send the <Amount></Amount> tags to do a partial RETURN. If this tag is not submitted or is not greater than 0, then the full amount of the original transaction is RETURNED.

```
<Request>
  <RequestType>ProcessSingleTransaction</RequestType>
  <ProcessType>RETURN</ProcessType>
  <PaymentType>CC</PaymentType>
  <TransactionID>100</TransactionID>
  <CreditCardNumber>1111</CreditCardNumber>
</Request>
```

Sample Request 3 – Partial Return on a previously settled transaction

```
<Request>
  <RequestType>ProcessSingleTransaction</RequestType>
  <ProcessType>RETURN</ProcessType>
  <PaymentType>CC</PaymentType>
  <Amount>2.50</Amount>
  <TransactionID>100</TransactionID>
  <CreditCardNumber>1111</CreditCardNumber>
</Request>
```

NOTE: Return on a previous settled transaction requires the transaction ID and the last 4 of the credit card number. If either of these are not included your transaction will not be processed.

Sample Request 4 – Force a previous AUTH ONLY transaction

```
<Request>
  <RequestType>ProcessSingleTransaction</RequestType>
  <ProcessType>PreviousSale</ProcessType>
  <TransactionID>100</TransactionID>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
TransactionType	(string)	DEBIT/CREDIT
TransactionDate	(datetime)	
Status	(string)	APPROVED/DECLINED/ERROR
StatusDescription	(string)	
ACHTransactionID	(string)	
ACHStatus	(string)	PENDING/RETURNED/SETTLED/TRANSMITTED
ACHStatusDescription	(string)	Log details of status. ie reason for return
Amount	(currency)	
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
EmailAddress	(string)	
Memo	(string)	
PaymentType	(string)	CC/EFT
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 Digits of the credit card number. Only returned if PaymentType = CC
ExpireMonth	(integer)	Only returned if PaymentType = CC
ExpireYear	(integer)	Only returned if PaymentType = CC
AuthorizationCode	(string)	
CVV2ResponseCode	(string)	
AVSResponseCode	(string)	
MaxScore	(string)	Maxmind Score
RecurrID	(integer)	RecurrID if RID was submitted
ErrorCode	(integer)	Declined Error Code
PartialPaymentID	(integer)	id assigned on original transaction
RequestedAmount	(currency)	
BALANCE	(currency)	Balance 1 left on Purchase/Debit card
CURRENCY	(string)	Currency 1 of Purchase/Debit card
BALLANCE2	(currency)	Balance 1 left on Purchase/Debit card
CURRENCY2	(string)	Currency 1 of Purchase/Debit card

If a PartialPaymentID is received if there was a partial authorization.

On partial authorizations the Amount is what was approved and the RequestedAmount is the amount submitted for the transaction.

Sample Response 1

```
<Result>
  <TransactionID>123456789</TransactionID>
  <TransactionType>CREDIT</TransactionType>
  <TransactionDate>2025-02-15</TransactionDate>
  <Status>APPROVED</Status>
  <StatusDescription>Transaction is APPROVED</StatusDescription>
  <Amount>150.00</Amount>
  <CustomerID>414000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <EmailAddress>John@home.com</EmailAddress>
  <Memo>6812300000</Memo>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>4111</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <AuthorizationCode>55468</AuthorizationCode>
  <CVV2ResponseCode/>
  <AVSResponseCode/>
</Result>
```

Sample Response 2 – Showing a Result2 option

```
<Result>
  <TransactionID>123456789</TransactionID>
  <TransactionType>CREDIT</TransactionType>
  <TransactionDate>2012-09-15</TransactionDate>
  <Status>APPROVED</Status>
  <Status2>Customer Updated In Vault</Status2>
  <StatusDescription>Transaction is APPROVED</StatusDescription>
  <Amount>150.00</Amount>
  <CustomerID>414000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <EmailAddress>john@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>111</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <AuthorizationCode>55468</AuthorizationCode>
  <CVV2ResponseCode/>
  <AVSResponseCode/>
</Result>
```

Partial Authorizations

What is Partial Authorization?

Partial authorizations occur when one of the cards above has less than the transaction amount. By using partial authorization a merchant can then ask for another form of payment for the remainder of the purchase.

How does it work?

- 1) The merchant submits the AUTH_ONLY, AUTH_CAPTURE or SALES request as they already do.
- 2) The gateway processes the request as normal and then replies with the message that a partial amount was approved due to that the balance on the card was not enough to cover the amount requested. The following will be supplied back to your software:
 - a. Amount Approved
 - b. Amount Requested
 - c. Partial Payment ID which is used to link subsequent transactions to the original request.
 - d. Balance Left On Card
 - e. Currency Of Card
 - f. Balance 2 Left On Card
 - g. Currency 2 of Card

NOTE: All transactions are put into a HOLD status until the final transaction is received that leaves a 0.00 amount due. At this point the system will automatically update these transactions to the original state that was requested.

NOTE 2: You can release the partial amounts associated for the Partial Payment ID manually if no final transaction is received by submitting a PREVIOUS_SALE ProcessType with the PartialPaymentID and the value of the original Partial Payment ID.

- 3) If you decide that you do not want your customer to submit other forms of payment or you do not wish to finish the partial approval still on HOLD then you should send a ProcessType of VOID and PartialPaymentID and the value of the original Partial Payment ID. DO NOT SEND THE TRANSACTION ID.
- 4) You can use as many different payment methods as needed to complete the order.

Do I have to use Partial Authorization?

No. For now this is an OPTED in option until the card industry makes it mandatory.

How do I opt in?

You can opt in 2 ways.

- 1) Change the Partial Authorization in the gateways Processing Settings to Yes.
- 2) Include the new field EnablePartialAuth with your request. This new field will over ride your Processing Settings.

Void Partial Auth

Void all partial auths that are still in HOLD status.

Request Parameters

Parameter Name	Required	Type	Notes
PartialPaymentID	Yes	(integer)	Original PartialPaymentID
ProcessType	Yes	(string)	VOID

Sample Request

```
<Request>  
  <RequestType>ProcessSingleTransaction</RequestType>  
  <PartialPaymentID>986523</PartialPaymentID>  
  <ProcessType>VOID</ProcessType>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
Status	(string)	VOIDED/ERROR
StatusDescription	(string)	

Sample Response

```
<Result>  
  <TransactionID>100123</TransactionID>  
  <Status>VOIDED</Status>  
  <StatusDescription>Transaction is VOIDED</StatusDescription>  
</Result>  
<Result>  
  <TransactionID>100124</TransactionID>  
  <Status>ERROR</Status>  
  <StatusDescription>Transaction has already been voided</StatusDescription>  
</Result>  
<Result>  
  <TransactionID>100125</TransactionID>  
  <Status>VOIDED</Status>  
  <StatusDescription>Transaction is VOIDED</StatusDescription>  
</Result>
```

Force Partial Auth

Force all partial auths that are still in HOLD status.

Request Parameters

Parameter Name	Required	Type	Notes
PartialPaymentID	Yes	(integer)	Original PartialPaymentID
ProcessType	Yes	(string)	PREVIOUS_SALE

Sample Request

```
<Request>  
  <RequestType>ProcessSingleTransaction</RequestType>  
  <PartialPaymentID>986523</PartialPaymentID>  
  <ProcessType>PREVIOUS_SALE</ProcessType>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
Status	(string)	APPROVED
StatusDescription	(string)	

Sample Response

```
<Result>  
  <TransactionID>100123</TransactionID>  
  <Status>APPROVED</Status>  
  <StatusDescription>Transaction has been forced.</StatusDescription>  
</Result>  
<Result>  
  <TransactionID>100124</TransactionID>  
  <Status>APPROVED</Status>  
  <StatusDescription> Transaction has been forced.</StatusDescription>  
</Result>  
<Result>  
  <TransactionID>100125</TransactionID>  
  <Status>APPROVED</Status>  
  <StatusDescription> Transaction has been forced.</StatusDescription>  
</Result>
```

Resubmit Transaction

ResubmitTransaction

Charge a customer from a previously APPROVED transaction.

Request Parameters

Parameter Name	Required	Type	Notes
TransactionID	Yes	(integer)	Original Transaction ID
Amount	Yes	(currency)	Payment amount
InvoiceNumber	No	(string)	Invoice number
InvoiceDescription	No	(string)	Invoice description

Sample Request

```
<Request>
  <RequestType>ResubmitTransaction</RequestType>
  <TransactionID>986523</TransactionID>
  <Amount>8.25</Amount>
  <InvoiceNumber>99</InvoiceNumber>
  <InvoiceDescription>Expedited Shipping</InvoiceDescription>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
TransactionID	(integer)	
TransactionType	(string)	DEBIT/CREDIT
TransactionDate	(datetime)	
Status	(string)	APPROVED/DECLINED/ERROR
StatusDescription	(string)	
ACHTransactionID	(string)	
ACHStatus	(string)	PENDING/RETURNED/SETTLED/TRANSMITTED
ACHStatusDescription	(string)	Log details of status. ie reason for return
Amount	(currency)	
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
EmailAddress	(string)	
Memo	(string)	
PaymentType	(string)	CC/EFT
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 Digits of the credit card number. Only returned if PaymentType = CC
ExpireMonth	(integer)	Only returned if PaymentType = CC
ExpireYear	(integer)	Only returned if PaymentType = CC
AuthorizationCode	(string)	
CVV2ResponseCode	(string)	
AVSResponseCode	(string)	
ErrorCode	(integer)	Declined Error Code

Sample Response

```
<Result>
  <TransactionID>123456789</TransactionID>
  <TransactionType>CREDIT</TransactionType>
  <TransactionDate>2025-02-15</TransactionDate>
  <Status>APPROVED</Status>
  <StatusDescription>Transaction is APPROVED.</StatusDescription>
  <ACHTransactionID/>
  <ACHStatus/>
  <ACHStatusDescription/>
  <Amount>8.25</Amount>
  <CustomerID>414000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Deer</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>65321</ZipCode>
  <EmailAddress>john@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <AuthorizationCode>55468</AuthorizationCode>
  <CVV2ResponseCode/>
  <AVSResponseCode>Y</AVSResponseCode>
</Result>
```


Request Listing – Vault

SearchVault

Used as a search engine to search for customers in the vault.

If no Request Parameter is supplied all customers are returned.

**** Not recommended if there are many customers as it may timeout!**

Request Parameters

Parameter Name	Required	Type	Notes
FirstName	No	(string)	Customer's first name
LastName	No	(string)	Customer's last name
Address	No	(string)	Customer's billing address
ZipCode	No	(string)	Customer's billing zip code
EmailAddress	No	(string)	Customer's email address
PaymentType	No	(string)	CC/EFT
CustomerID	No	(string)	

Sample Request

```
<Request>
  <RequestType>SearchVault</RequestType>
  <FirstName/>
  <LastName>Doe</LastName>
  <Address/>
  <ZipCode/>
  <EmailAddress/>
  <PhoneNumber/>
  <PaymentType/>
</Request>
```

Response Parameters

Parameter Name	Type
CustomerID	(string)
FirstName	(string)
LastName	(string)

Sample Response

```
<Result>
  <CustomerID>4041000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
</Result>
```

ShowVaultDetails

Used as a search engine to search for customers in the vault.

Request Parameters

Parameter Name	Required	Type	Notes
CustomerID	Yes	(string)	

Sample Request

```
<Request>  
  <RequestType>ShowVaultDetails</RequestType>  
  <CustomerID>4041000000</CustomerID>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
City	(string)	
State	(string)	
ZipCode	(string)	
EmailAddress	(string)	
PhoneNumber	(string)	
PaymentType	(string)	
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 digits of credit card number
ExpireMonth	(string)	
ExpireYear	(string)	
ABANumber	(string)	
AccountNumber	(string)	Last 4 digits of bank account number
EFTType	(string)	PPD, CCD, WEB, TEL – If applicable

Sample Response

```
<Result>  
  <CustomerID>4041000000</CustomerID>  
  <FirstName>John</FirstName>  
  <LastName>Doe</LastName>  
  <Address>1706 W. 182nd Street</Address>  
  <City>Chesapeake</City>  
  <State>VA</State>  
  <ZipCode>23320</ZipCode>  
  <EmailAddress>John@home.com</EmailAddress>  
  <PhoneNumber>888-555-1212</PhoneNumber>  
  <PaymentType>CC</PayType>  
  <CardType>VI</CardType>  
  <CreditCardNumber>5454</CreditCardNumber>  
  <ExpireMonth>12</ExpireMonth>  
  <ExpireYear>2025</ExpireYear>  
  <ABANumber/>  
  <AccountNumber/>  
</Result>
```

Request Listing – Customers

CustomerAdd

* Adds a new Customer to the Vault.

Request Parameters

Parameter Name	Type	Required	Notes
CustomerID	(string)	Yes	A unique identifier for the customer
FirstName	(string)	Special	Required if PaymentType = EFT
LastName	(string)	Yes	
Address	(string)	Yes	
City	(string)	Special	Required if PaymentType = EFT
State	(string)	Special	Required if PaymentType = EFT
ZipCode	(string)	Yes	
PhoneNumber	(string)	Special	Required if PaymentType = EFT
EmailAddress	(string)	No	Required to send payment receipts
PaymentType	(string)	Yes	
CreditCardNumber	(string)	Special	Required if PaymentType = CC
CVV2	(string)	No	
ExpireMonth	(string)	Special	Required if PaymentType = CC
ExpireYear	(string)	Special	Required if PaymentType = CC
ABANumber	(string)	Special	Required if PaymentType = EFT
AccountNumber	(string)	Special	Required if PaymentType = EFT
EFTType	(string)	Special	Required if PaymentType = EFT PPD/CCD/WEB/TEL

* No authorization is done to the customer's form of payment.

* If PaymentType is CC the expiration date will be checked to see that it is greater than current year month.

Sample Request

```
<Request>
  <RequestType>CustomerAdd</RequestType>
  <CustomerID>4041000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <PhoneNumber>360-888-6236</PhoneNumber>
  <EmailAddress>John@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>5454545454545454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <ABANumber/>
  <AccountNumber/>
  <EFTType/>
</Request>
```

Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
PaymentType	(string)	
CreditCardNumber	(string)	Last 4 Digits of CreditCardNumber PaymentType = CC
ExpireMonth	(string)	
ExpireYear	(string)	
ABANumber	(string)	
AccountNumber	(string)	Last 4 Digits of AccountNumber PaymentType = EFT
EFTType	(string)	PPD/CCD/WEB/TEL

Sample Response

```
<Result>
  <RequestType>CustomerAdd</RequestType>
  <CustomerID>4041000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <PhoneNumber>888-555-1212</PhoneNumber>
  <EmailAddress>John@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>5454545454545454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <ABANumber />
  <AccountNumber />
  <EFTType />
</Result>
```

CustomerUpdate

Update a current Vault Customers information.

Request Parameters

Parameter Name	Type	Required	Notes
CustomerID	(string)	Yes	A unique identifier for the customer
FirstName	(string)	No	
LastName	(string)	Yes	
Address	(string)	Yes	
ZipCode	(string)	Yes	
PaymentType	(string)	Yes	
CreditCardNumber	(string)	Special	If Updating Card Or Expiration
ExpirationExpireMonth	(string)	Special	Required if PaymentType = CC
ExpireYear	(string)	Special	Required if PaymentType = CC
ABANumber	(string)	Special	Required if PaymentType = EFT
AccountNumber	(string)	Special	Required if PaymentType = EFT
EFTType	(string)	Special	Required if PaymentType = EFT PPD/CCD/WEB/TEL
PhoneNumber	(string)	Special	Required if PaymentType = EFT
City	(string)	Special	Required if PaymentType = EFT
State	(string)	Special	Required if PaymentType = EFT

Sample Request

```
<Request>  
  <RequestType>CustomerUpdate</RequestType>  
  <CustomerID>4041000000</CustomerID>  
  <FirstName>John</FirstName>  
  <LastName>Doe</LastName>  
  <Address>1706 W. 182nd Street</Address>  
  <ZipCode>9999</ZipCode>  
  <PaymentType>CC</PaymentType>  
  <CreditCardNumber>5454545454545454</CreditCardNumber>  
  <ExpireMonth>12</ExpireMonth>  
  <ExpireYear>2020</ExpireYear>  
  <ABANumber/>  
  <AccountNumber/>  
  <EFTType/>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
PaymentType	(string)	
CreditCardNumber	(string)	Last 4 Digits of CreditCardNumber PaymentType = CC
CVV2	(string)	
ExpireMonth	(string)	
ExpireYear	(string)	
ABANumber	(string)	
AccountNumber	(string)	Last 4 Digits of AccountNumber PaymentType = EFT

Sample Response

```
<Result>
  <RequestType>CustomerUpdate</RequestType>
  <CustomerID>4041000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <PhoneNumber>888-555-1212</PhoneNumber>
  <EmailAddress>John@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
</Result>
```

CustomerUpdatePayOnly

* Updates the credit card number and expiration date or updates an EFT routing and account number.

Request Parameters

Parameter Name	Type	Required	Notes
CustomerID	(string)	Yes	A unique identifier for the customer
PaymentType	(string)	Yes	
CreditCardNumber	(string)	Special	Required if PaymentType = CC
ExpireMonth	(string)	Special	Required if PaymentType = CC
ExpireYear	(string)	Special	Required if PaymentType = CC
ABANumber	(string)	Special	Required if PaymentType = EFT
AccountNumber	(string)	Special	Required if PaymentType = EFT
EFTType	(string)	Special	Required if PaymentType = EFT

Sample Request

```
<Request>
  <RequestType>CustomerUpdatePayOnly</RequestType>
  <CustomerID>4041000000</CustomerID>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>4111111111111111</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2025</ExpireYear>
</Request>
```

Sample Response

```
<Result>
  <RequestType>CustomerUpdatePayOnly</RequestType>
  <CustomerID>4041000000</CustomerID>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>1111</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2025</ExpireYear>
</Result>
```

AddUpdateCustomer

* Adds a new Customer to the Vault or Updates current customer.

Request Parameters

Parameter Name	Type	Required	Notes
CustomerID	(string)	Yes	A unique identifier for the customer
FirstName	(string)	Special	Required if PaymentType = EFT
LastName	(string)	Yes	
Address	(string)	Yes	
City	(string)	Special	Required if PaymentType = EFT
State	(string)	Special	Required if PaymentType = EFT
ZipCode	(string)	Yes	
PhoneNumber	(string)	Special	Required if PaymentType = EFT
EmailAddress	(string)	No	Required to send payment receipts
PaymentType	(string)	Yes	
CreditCardNumber	(string)	Special	Required if PaymentType = CC
ExpireMonth	(string)	Special	Required if PaymentType = CC
ExpireYear	(string)	Special	Required if PaymentType = CC
ABANumber	(string)	Special	Required if PaymentType = EFT
AccountNumber	(string)	Special	Required if PaymentType = EFT
EFTType	(string)	Special	Required if PaymentType = EFT PPD/CCD/WEB/TEL
RequestType2	(string)	No	TestBeforeVaultUpdateInsert

A unique feature to this function is

```
<RequestType2>TestBeforeVaultUpdateInsert</RequestType2>
```

Add this along with the normal RequestType and an auth for 1 cent or what is sent in the Amount will be performed before the customer is added or updated. **NOTE: Processor Per Transaction Fees Apply**

* If PaymentType is CC the expiration date will be checked to see that it is greater than current year month.

Sample Request

```
<Request>  
  <RequestType>AddUpdateCustomer</RequestType>  
  <CustomerID>4041000000</CustomerID>  
  <FirstName>John</FirstName>  
  <LastName>Doe</LastName>  
  <Address>1706 W. 182nd Street</Address>  
  <ZipCode>23320</ZipCode>  
  <PhoneNumber>360-888-6236</PhoneNumber>  
  <EmailAddress>John@home.com</EmailAddress>  
  <PaymentType>CC</PaymentType>  
  <CreditCardNumber>5454545454545454</CreditCardNumber>  
  <ExpireMonth>12</ExpireMonth>  
  <ExpireYear>2020</ExpireYear>  
  <ABANumber/>  
  <AccountNumber/>  
  <EFTType/>  
</Request>
```


Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
ZipCode	(string)	
PaymentType	(string)	
CreditCardNumber	(string)	Last 4 Digits of CreditCardNumber PaymentType = CC
ExpireMonth	(string)	
ExpireYear	(string)	
ABANumber	(string)	
AccountNumber	(string)	Last 4 Digits of AccountNumber PaymentType = EFT
EFTType	(string)	PPD/CCD/WEB/TEL if applicable

Sample Response

```
<Result>
  <RequestType>CustomerAdd</RequestType>
  <CustomerID>4041000000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <ZipCode>23320</ZipCode>
  <PhoneNumber>888-555-1212</PhoneNumber>
  <EmailAddress>John@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>5454545454545454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <ABANumber />
  <AccountNumber />
  <EFTType />
</Result>
```

CustomerRemove

Remove a current Vault Customers information.

Request Parameters

Parameter Name	Type	Required	Notes
CustomerID	(string)	Yes	A unique identifier for the customer

Sample Request

Sample Request

```
<Request>  
  <RequestType>CustomerRemove</RequestType>  
  <CustomerID>4041000000</CustomerID>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	

Sample Response

```
<Result>  
  <CustomerID>4041000000</CustomerID>  
  <Status>Success</Status>  
  <StatusDescription>Customer Was Removed.</StatusDescription>  
</Result>
```

ExpiringVaultCustomers

Search for Vault Customers who's credit card is expiring.

Request Parameters

Parameter Name	Type	Required	Notes
NumMonths	(number)	Yes	1 – Current Month 2 – This Month Plus 1 3 – This Month Plus 2 Etc.

Sample Request

```
<Request>  
  <RequestType>ExpiringVaultCustomers</RequestType>  
  <NumMonths>1</NumMonths>  
</Request>
```

Sample Response

```
<ResultDetails>  
  <ExpirationDate>2020-05</ExpirationDate>  
  <SubResultCount>2</SubResultCount>  
  <Result>  
    <CustomerID>122</CustomerID>  
  </Result>  
  <Result>  
    <CustomerID>123</CustomerID>  
  </Result>  
</ResultDetails>  
<ResultDetails>  
  <ExpirationDate>2020-06</ExpirationDate>  
  <SubResultCount>2</SubResultCount>  
  <Result>  
    <CustomerID>124</CustomerID>  
  </Result>  
  <Result>  
    <CustomerID>125</CustomerID>  
  </Result>  
</ResultDetails>
```

ExpiringRecurringCustomers

Search for Recurring Customers who's credit card is expiring.

Request Parameters

Parameter Name	Type	Required	Notes
NumMonths	(number)	Yes	1 – Current Month 2 – This Month Plus 1 3 – This Month Plus 2 Etc.

Sample Request

```
<Request>  
  <RequestType>ExpiringRecurringCustomers</RequestType>  
  <NumMonths>1</NumMonths>  
</Request>
```

Sample Response

```
<ResultDetails>  
  <ExpirationDate>2025-05</ExpirationDate>  
  <SubResultCount>2</SubResultCount>  
  <Result>  
    <RecurrID>122</RecurrID>  
    <BillingName>John Doe</BillingName>  
    <PhoneNumber>8885551212</PhoneNumber>  
    <EmailAddress>home@home.com</EmailAddress>  
  </Result>  
  <Result>  
    <RecurrID>123</RecurrID>  
    <BillingName>John Doe 2</BillingName>  
    <PhoneNumber>8885551212</PhoneNumber>  
    <EmailAddress>home2@home.com</EmailAddress>  
  </Result>  
</ResultDetails>  
<ResultDetails>  
  <ExpirationDate>2025-06</ExpirationDate>  
  <SubResultCount>2</SubResultCount>  
  <Result>  
    <RecurrID>124</RecurrID>  
    <BillingName>John Doe 3</BillingName>  
    <PhoneNumber>8885551212</PhoneNumber>  
    <EmailAddress>home3@home.com</EmailAddress>  
  </Result>  
  <Result>  
    <RecurrID>125</RecurrID>  
    <BillingName>John Doe 4</BillingName>  
    <PhoneNumber>8885551212</PhoneNumber>  
    <EmailAddress>home4@home.com</EmailAddress>  
  </Result>  
</ResultDetails>
```

SearchRecurring

Search for Recurring Entries.

Request Parameters

Parameter Name	Type	Required	Notes
PaymentType	(string)	No	
Amount	(currency)	No	
BillingName	(string)	No	
Address	(string)	No	
City	(string)	No	
State	(string)	No	
ZipCode	(string)	No	
Country	(string)	No	
PhoneNumber	(string)	No	
EmailAddress	(string)	No	
CustomerID	(string)	No	
Status	(string)	No	
RID	(integer)	No	Recurring Recipe ID

Sample Request

```
<Request>  
  <RequestType>SearchRecurring</RequestType>  
  <Amount>10.00</Amount>  
  <State>PA</State>  
  <Status>Active</Status>  
</Request>
```

Sample Response

```
<Result>  
  <RecurrID>505</RecurrID>  
  <Status>Active</Status>  
  <BillingName>John Brown</BillingName>  
</Result>  
<Result>  
  <RecurrID>585</RecurrID>  
  <Status>Active</Status>  
  <BillingName>Jane Doe</BillingName>  
</Result>
```

ShowRecurringCustomer

Search for Recurring Entries.

Request Parameters

Parameter Name	Type	Required	Notes
RecurrID	(string)	Yes	Value from Search Recurring

Sample Request

```
<Request>
  <RequestType>ShowRecurringCustomer</RequestType>
  <RecurrID>585</RecurrID>
</Request>
```

Sample Response

```
<Result>
  <RecurrID>585</RecurrID>
  <Status>Active</Status>
  <CustomerID>998567388</CustomerID>
  <BillingName>Jane Doe</BillingName>
  <Address>1313 15th St</Address>
  <City>Beverly Hills</City>
  <State>CA</State>
  <ZipCode>90210</ZipCode>
  <Country>USA</Country>
  <PhoneNumber>555-555-5555</PhoneNumber>
  <EmailAddress>jane@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>*****5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
</Result>
```

UpdateRecurringCustomer

Update a Recurring Customer

Request Parameters

Parameter Name	Type	Required	Notes
RecurrID	(string)	Yes	Value from Search Recurring
CustomerID	(string)	Yes	
BillingName	(string)	Yes	
Address	(string)	Yes	
City	(string)	Yes	
State	(string)	Yes	
ZipCode	(string)	Yes	
Country	(string)	Yes	
PhoneNumber	(string)	Yes	
EmailAddress	(string)	Yes	
Amount	(currency)	No	
CreditCardNumber	(string)	No	
ExpireMonth	(string)	No	Yes if updating Credit Card
ExpireYear	(string)	No	Yes if updating Credit Card
PaymentType	(string)	No	Yes if updating Credit Card
RecurStatus	(string)	No	Active/Suspended/Canceled

Sample Request

```
<Result>
  <RecurrID>585</RecurrID>
  <BillingName>Jane Doe</BillingName>
  <Address>1313 15th St</Address>
  <City>Beverly Hills</City>
  <State>CA</State>
  <ZipCode>90210</ZipCode>
  <Country>USA</Country>
  <PhoneNumber>555-555-5555</PhoneNumber>
  <EmailAddress>jane@home.com</EmailAddress>
  <PaymentType>CC</PaymentType>
  <CreditCardNumber>5454545454545454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
</Result>
```

Sample Response

```
<ResponseSummary>
  <RequestType>UpdateRecurringCustomer</RequestType>
  <Status>Success</Status>
  <StatusDescription>Request was successful.</StatusDescription>
  <ResultCount>1</ResultCount>
  <TimeStamp>2025-07-12 09:05:58</TimeStamp>
</ResponseSummary>
<Result>
  <RecurrID>585</RecurrID>
  <Status>Customer Details updated. Customer Payment information Updated</Status>
</Result>
```

Please note that an update will **always** update CustomerID,BillingName,Address,City,State,ZipCode,Country,PhoneNumber and EmailAddress
Amount will only be updated if a value is submitted.
Payment information will only be updated if the values are submitted

RecurStatus will only update if it is submitted.

Vault To Recurring

Create Recurring Transactions From Vault

Used to create an individual Debit or Credit Transaction from Vault Customers

Request Parameters

Parameter Name	Required	Type	Notes
CustomerID	Yes	(string)	Identifies the vault customer
RecurAmount	Yes	(currency)	
RID	Yes	(integer)	Valid Recipe ID to use
RecurDay	No	(integer)	Day of week or month. IE Wednesday or 05 or 15
RecurTimes	No	(integer)	Times to recur. 0 for unlimited

Sample Request

```
<Request>  
  <RequestType>VaultCreateRecurring</RequestType>  
  <CustomerID>40441000</CustomerID>  
  <RecurAmount>15.00</RecurAmount>  
  <RID>1</RID>  
  <RecurDay>05</RecurDay>  
</Request>
```

Response Parameters

Parameter Name	Type	Notes
CustomerID	(string)	
FirstName	(string)	
LastName	(string)	
Address	(string)	
City	(string)	
State	(string)	
ZipCode	(string)	
EmailAddress	(string)	
Phone	(string)	
PaymentType	(string)	CC/EFT
CardType	(string)	MC/VI/DS/AE/DC/JC
CreditCardNumber	(string)	Last 4 Digits of the credit card number. Only returned if PaymentType = CC
ExpireMonth	(integer)	Only returned if PaymentType = CC
ExpireYear	(integer)	Only returned if PaymentType = CC
ABANumber	(string)	Only returned if PaymentType = EFT
AccountNumber	(string)	Last 4 Digits of the bank account number. Only returned if PaymentType = EFT
RecurrID	(integer)	Unique Recurring ID

Sample Response

```
<Result>
  <CustomerID>40441000</CustomerID>
  <FirstName>John</FirstName>
  <LastName>Doe</LastName>
  <Address>1706 W. 182nd Street</Address>
  <City>Hershey</City>
  <State>PA</State>
  <ZipCode>99764</ZipCode>
  <EmailAddress>John@home.com</EmailAddress>
  <Phone>8885863346</Phone>
  <PaymentType>CC</PaymentType>
  <CardType>VI</CardType>
  <CreditCardNumber>5454</CreditCardNumber>
  <ExpireMonth>12</ExpireMonth>
  <ExpireYear>2020</ExpireYear>
  <RecurrID>55</RecurrID>
</Result>
```

Maxmind Definitions

distance - Distance from IP address to Billing Location in kilometers (large distance = higher risk)

countryMatch - Whether country of IP address matches billing address country (mismatch = higher risk)

countryCode - Country Code of the IP address

highRiskCountry - Whether IP address or billing address country is in Egypt, Ghana, Indonesia, Lebanon, Macedonia, Morocco, Nigeria, Pakistan, Romania, Serbia and Montenegro, Ukraine, or Vietnam.

ip_region - Estimated State/Region of the IP address, ISO-3166-2 / FIPS 10-4 code

ip_city - Estimated City of the IP address

ip_latitude - Estimated Latitude of the IP address

ip_longitude - Estimated Longitude of the IP address

ip_isp - ISP of the IP address

ip_org - Organization of the IP address

anonymousProxy - Whether IP address is Anonymous Proxy (anonymous proxy = very high risk)

proxyScore - Likelihood of IP Address being an Open Proxy

isTransProxy - Whether IP address is in our database of known transparent proxy servers, returned if forwardedIP is passed as an input.

freeMail - Whether e-mail is from free e-mail provider (free e-mail = higher risk)

carderEmail - Whether e-mail is in database of high risk e-mails

binMatch - Whether country of issuing bank based on BIN number matches billing address country

binCountry - Country Code of the bank which issued the credit card based on BIN number

binNameMatch - Whether name of issuing bank matches inputted binName. A return value of Yes provides a positive indication that cardholder is in possession of credit card.

binName - Name of the bank which issued the credit card based on BIN number*. Available for approximately 96% of BIN numbers

binPhoneMatch - Whether customer service phone number matches inputted binPhone. A return value of Yes provides a positive indication that cardholder is in possession of credit card.

binPhone - Customer service phone number listed on back of credit card*. Available for approximately 75% of BIN numbers. In some cases phone number returned may be outdated.

custPhoneInBillingLoc - Whether the customer phone number is in the billing zip code. A return value of Yes provides a positive indication that the phone number listed belongs to the cardholder. A return value of No indicates that the phone number may be in a different area, or may not be listed in our database. Currently we only support US Phone numbers.

shipForward - Whether shipping address is in database of known mail drops

cityPostalMatch - Whether billing city and state match zipcode. Currently available for US addresses only.

shipCityPostalMatch - Whether shipping city and state match zipcode. Currently available for US addresses only

score - Overall fraud score based on outputs listed above. This is the original fraud score, and is based on a simple formula. It has been replaced with riskScore (see below), but is kept for backwards compatibility.

explanation - A brief explanation of the score, detailing what factors contributed to it, according to maxmind formula.

riskScore - New fraud score representing the estimated probability that the order is fraud, based off of analysis of past minFraud transactions.

err - Returns an error string with a warning message or a reason why the request failed.

Sample PeopleCode

Sample Send Synchronous Message

```
/* *****  
* Sample PeopleCode for sending Integration Broker Messages  
* By John Doe  
* *****/  
  
Local Message &RequestMessage, &ResponseMessage;  
Local Rowset &CustomerTable;  
  
&RequestMessage = CreateMessage(Message.<YourMessageName>);  
  
/* Check to see if the message is active before trying to send*/  
If (&RequestMessage.IsActive) Then  
  
    /* Instantiate Rowset */  
    &CustomerTable = CreateRowset(Record.<YourCustomerTable>);  
  
    /* Fill the rowset with customers */  
    &CustomerTable.Fill("Where <Search Field>=:1",<Variable or static criteria>);  
  
    /* Check to see if we have any customers that met our criteria */  
    If (&CustomerTable.RowCount > 0) Then  
  
        /* Copy the matching fields data to the request message */  
        &RequestMessage.CopyRowset(&CustomerTable);  
  
        /* Send a synchronous message and get the response */  
        &ResponseMessage = &RequestMessage.SyncRequest();  
  
    End-If;  
  
End-If;
```

Sample PHP Code

Sample CURL POST

```
<?php
/*****
 * Post to Quantum Gateway using CURL
 * By John Doe
 *****/

$url = "https://secure.quantumgateway.com/cgi/xml_requester.php";

$xml = "<QGWRequest>\n";
$xml .= "  <Authentication>\n";
$xml .= "    <GatewayLogin>99</GatewayLogin>\n";
$xml .= "    <GatewayKey>CKhK29CkVrmHUbJ</GatewayKey>\n";
$xml .= "  </Authentication>\n";
$xml .= "  <Request>\n";
$xml .= "    <RequestType>ShowVaultDetails</RequestType>\n";
$xml .= "    <CustomerID>404100000</CustomerID>\n";
$xml .= "  </Request>\n";
$xml .= "</QGWRequest>\n";

$header = "POST $url HTTP/1.1\n";
$header.= "Host: <your domain>\n";
$header.= "Content-Length: ".strlen($xml)."\n";
$header.= "Content-type: text/xml; charset=UTF8\n";
$header.= "Connection: close; Keep-Alive\n\n";
$header.= $xml;

$ch = curl_init();

curl_setopt($ch, CURLOPT_URL,$this->gatewayUrl);
curl_setopt($ch, CURLOPT_TIMEOUT, 120);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
curl_setopt($ch, CURLOPT_HTTPHEADER, $header);
$responseXml=curl_exec($ch);

curl_close($ch);

?>
```

Sample Xml Parsing

```

<?php
/*****
* Parse Xml String Using the DOMDocument Object
* By John Doe
*****/

$xml = "<QGWRequest>\n";
$xml .= "    <ResponseSummary>\n";
$xml .= "        <RequestType>SearchVault</RequestType>\n";
$xml .= "        <Status>Success</Status>\n";
$xml .= "        <StatusDescription>Request was successful.</StatusDescription>\n";
$xml .= "        <TimeStamp>2/2/2007 8:34:28 AM</TimeStamp>\n";
$xml .= "    </ResponseSummary>\n";
$xml .= "    <Result>\n";
$xml .= "        <CustomerID>404100000</CustomerID>\n";
$xml .= "    </Result>\n";
$xml .= "</QGWRequest>\n";

/* Instantiate the DOMDocument */
$xmlDoc = new DOMDocument;

/* Load the xml into the DOMDocument */
$xmlDoc->LoadXml($xml);

/* Get a NodeList for Result */
/* We could also parse the ResponseSummary nodes by changing Result */
/* ResponseSummary */
$Elements = $xmlDoc->GetElementsByTagName("Result");

/* Loop through Result Nodes */
$i=0;
foreach($Result_nodeList as $Element){
    /* Get a NodeList for everything inside the Result nodes */
    $Elements = $Element->getElementsByTagName("*");

    /* Loop through all inside each Result node */
    foreach($Elements as $node){
        /* Put the tagname in $nodeName */
        $nodeName = $node->nodeName;
        /* Put the node value in $nodeValue */
        $nodeValue = $node->nodeValue;

        /* Create an array of the node tags and values */
        $responseArray[$i][$nodeName] = $nodeValue;
    }
    $i++;
}
/* Now use your database handler to store/update/select */
foreach($responseArray as $data){

    /*Example insert*/
    $query = "insert into <tablename>(<field>) values('$data[CustomerID]')";

    /*Example select*/
    $query = "select * from <tablename> where <field> = '$data[CustomerID]'";
}

?>

```

Request Errors

<Status>ERROR</Status>

All RequestTypes

Unapproved IPAddress

AddUpdateCustomer

gwLogin And Vault Key Dont Match

Credit Card Failed

Missing ABANumber

Missing AccountNumber

Missing ABANumber MissingAccountNumber

ExpireYear is before this year.

ExpireMonth is this month or has expired.

ExpireYear is before this year. ExpireMonth is this month or has expired.

CreateTransaction

ALL

gwLogin And Vault Key Dont Match!

Record already exists for the above information

NO CUSTOMERID

EFT Only

EFTType was not submitted

The phone number is either not present or it is less than 10 digits.

Billing Name Is Not Present.

Billing Address Is Not Present.

Billing City Is Not Present.

Billing State Is Not Present.

Billing ZipCode Is Not Present.

CustomerAdd

GatewayLogin And GatewayKey Dont Match!

CustomerID already exists.

Any of the 3 below or in any combination

Missing ABANumber

Missing AccountNumber

Missing EFTType

Any of the 6 below or in any combination

Missing CreditCardNumber.

CreditCardNumber is too short.

Missing ExpireMonth.

Missing ExpireYear.

ExpireYear is before this year.

ExpireMonth is this month or has expired.

CustomerRemove

GatewayLogin And GatewayKey Dont Match

Customer Not Found

CustomerUpdate

GatewayLogin And GatewayKey Dont Match

Customer Not Found

Any of the 3 below or in any combination

Missing ABANumber

Missing AccountNumber

Missing EFTType

Any of the 6 below or in any combination

Missing CreditCardNumber.

CreditCardNumber is too short.

Missing ExpireMonth.

Missing ExpireYear.

ExpireYear is before this year.

ExpireMonth is this month or has expired.

ExpiringVaultCustomers

GatewayLogin And GatewayKey Dont Match

ProcessSingleTransaction

ALL ProcessType

RestrictKey and Gateway Account Do Not Match

Test Cards Not Enabled.

CC Only

Invalid Credit Card Number

EFT Only

EFTType was not submitted

The phone number is either not present or it is less than 10 digits.

Billing Name Is Not Present.

Billing Address Is Not Present.

Billing City Is Not Present.

Billing State Is Not Present.

Billing ZipCode Is Not Present.

ProcessType: RETURN – From Previously Submitted Transaction

Transaction Does Not Exist

Last 4 of Credit Card was not submitted.

Last 4 of Credit Card does match the transaction.

Payment Information Has Been Removed. Transaction can not be refunded.

Transaction has already been voided.

ProcessType: VOID

TransactionID not found

Payment Information Has Been Removed. Transaction can not be voided.

Transaction has already been voided

Transaction has been batched out and can not be voided.

ProcessType: PREVIOUS_SALE

TransactionID not found

Payment Information Has Been Removed. Transaction can not be forced.

Transaction has already been voided

Transaction has been batched out and can not be forced.

ProcessType: VOID - Partial Auth

PartialPaymentID not found or the transaction has already been voided or forced.

Payment Information Has Been Removed. Transactions can not be voided.
Transaction has already been voided

ProcessType: PREVIOUS_SALE - Force Partial Auth

PartialPaymentID not found or it has already been voided or forced.

ResubmitTransaction

RestrictKey and Gateway Account Do Not Match
TransactionID not found
Amount not found.
Payment Information Has Been Removed. Transaction can not be Resubmitted.
Original Transaction Was Not Approved.

SearchRecurring

RestrictKey and Gateway Account Do Not Match

SearchTransactions

RestrictKey and Gateway Account Do Not Match
No Search Criteria Found

SearchVault

GatewayLogin And GatewayKey Dont Match
No Results Found

ShowRecurringCustomer

RestrictKey and Gateway Account Do Not Match
No Such Recurring Customer

ShowTransactionDetails

GatewayLogin And GatewayKey Dont Match
Transaction not found.

ShowVaultDetails

GatewayLogin And GatewayKey Dont Match
No CustomerID Found
CustomerID Not Found

UpdateRecurringCustomer

RestrictKey and Gateway Account Do Not Match
No Such Recurring Customer

VaultCheck

gwLogin And Vault Key Dont Match

VaultCreateRecurring

GatewayLogin And GatewayKey Dont Match
No CustomerID Found
No RID Found
RID is not valid
CustomerID Not Found
RecurDay is not valid for this recipe type

Field Index

A

ABANumber 10, 12, 14, 16, 19, 20, 21, 22, 23
AccountNumber 10, 12, 14, 16, 19, 20, 21, 22, 23
AccountType 14
ACHStatus 9, 10, 12, 16
ACHStatusDescription 10, 12, 16
ACHTransactionID 9, 10, 12, 16
Address 10, 12, 14, 16, 18, 19, 20, 21, 22, 23
Amount 10, 12, 14, 16
AuthorizationCode 10, 12, 16
AVSResponseCode 10, 12, 16

C

CardType 10, 12, 16, 19
City 14
Country 14
CreditCardNumber 10, 12, 14, 16, 19, 20, 21, 22, 23
CustomerID 8, 10, 12, 16, 18, 19, 20, 21, 22, 23
CVV2 14, 20, 21, 22, 23
CVV2ResponseCode 10, 12, 16

E

EFTType 20
EmailAddress 10, 12, 14, 16, 18, 19, 20
ExpireMonth 10, 12, 14, 16, 19, 20, 21, 22, 23
ExpireYear 10, 12, 14, 16, 19, 20, 21, 22, 23

F

FirstName 10, 12, 14, 16, 18, 19, 20, 21, 22, 23

G

GatewayKey 4, 7
GatewayLogin 4, 7

I

InvoiceDescription 14
InvoiceNumber 14

IPAddress 14

L

LastName 10, 12, 14, 16, 18, 19, 20, 21, 22, 23

M

Memo 10, 12, 14, 16

P

PaymentType ...8, 10, 12, 14, 16, 18, 19, 20, 21, 22, 23
PhoneNumber 14, 18, 19, 20

R

ResultCount 7

S

SettlementBatchID 8, 10, 12, 16
SettlementDate 10, 12, 16
SettlementDateBegin 8
SettlementDateEnd 8
State 14
Status 7, 8, 9, 10, 12, 16
StatusDescription 7, 10, 12, 16

T

TimeStamp 7
TransactionDate 9, 10, 12, 14, 16
TransactionDateBegin 8
TransactionDateEnd 8
TransactionID 9, 10, 12, 16
TransactionType 7, 8, 10, 12, 14, 16

Z

ZipCode 10, 12, 14, 16, 18, 19, 20, 21, 22, 23