ProtectPay® API Appendix

Version 4

Appendix A: Response Elements

A 'ResultCode' element is returned at both the request level and at the transaction level for all requests.
The ResultCode of the request indicates the outcome of the request.
The ResultCode of the transaction element indicates WHY a certain response was returned.
For example, the RequestResult.ResultCode and ResultMessage may indicate a 201 - Invalid Argument Error has occurred, while the Transaction.ResultCode.ResultCode and ResultMessage will contain additional info such as 'Invalid ExpDate'.

A.1 ProtectPay API Request Response Values

The following response codes are returned in the [RequestResult] object. They are generated by ProtectPay and returned as the status of the API Request. Response codes other than '00' indicate that ProtectPay was unable to submit a transaction to the merchant processor.

ProtectPay API Request Response Values

Code	Message
00	Success.
300	Authentication error.
301	Invalid argument error. *Error details returned in Transaction.ResultCode.ResultMessage.
302	Invalid invoice number.
303	Gateway Timeout Error
304	System of record account error
305	Invalid track data.
306	Unsupported error
307	Internal system error. *Error details returned in Transaction.ResultCode.ResultMessage.
308	Invalid credit card
309	Insufficient payment methods
310	Unsupported currency code
311	Invalid argument error. *Error details returned in Transaction.ResultCode.ResultMessage.
312	Address validation error
313	ID validation error
314	Account validation error
315	Payment Method validation error
316	Call failed for an unspecified reason
317	Duplicate Account Number Found
318	Country code not supported
319	Argument format error
320	Argument required error
321	Invalid password
322	Latest EULA not signed
326	Invalid track data
330	Authorization Error
341	Payment method does not exist
345	Unable to process your request
346	Not subscribed to AutoUpdater
347	Not enrolled to auto update card brand
348	Transaction successfully voided. *Auto-Void Feature
349	Transaction void failed. *Auto-Void Feature

700	Invalid payment method ID

A.2 Processor Response Values

The following response codes are returned in the [RequestResult] object. Response codes other than '00' indicate that ProtectPay was able to successfully submit a transaction to the merchant processor and the processor failed and/or refused to pass the transaction to the issuer.

ProtectPay API Request Processor Response Values

Code	Message
200	Gateway authentication error
201	Gateway invalid argument error *Error details returned in Transaction.ResultCode.ResultMessage.
204	Gateway account status error *Error details returned in Transaction.ResultCode.ResultMessage.
206	Gateway unsupported transaction request *Error details returned in Transaction.ResultCode.ResultMessage.
207	Gateway Internal system error *Error details returned in Transaction.ResultCode.ResultMessage.
212	Gateway Address validation error. *Error details returned in Transaction.ResultCode.ResultMessage.
214	Gateway Invalid Destination Account
223	Gateway Duplicate transaction
224	Gateway Amount exceeds single transaction limit
225	Gateway Amount exceeds monthly volume limit
226	Gateway Invalid track 1
227	Gateway reported decline based on user settings
230	Unauthorized service requested on Gateway
236	Capture amount exceeds allowed amount
237	MCC doesn't allow capturing for a greater amount
250	CVV code no match (transaction reversed)

A.3 ProPay® Processor Specific Response Values

The following response codes are returned in the [RequestResult] object. These response codes only apply if ProPay is the processor.

ProtectPay API Request ProPay Processor Response Values

Code	Message	Transaction Status
542	Invalid receiving email	Error
544	Invalid amount	Error
551	Invalid trans num or unable to act due to funding	Decline
561	Amount exceeds single transaction limit	Decline
562	Amount exceeds monthly volume limit	Decline
567	Unauthorized service requested	Error
568	Account not affiliated	Decline

A.4 Issuer Response Values

The following response codes are returned in the [Transaction.RequestResult] object. The following table details the responses from the transaction request as returned by the issuer. They indicate that the request was successfully submitted to the processor, and the code and reason are indications of the success or failure as returned by the card-issuing financial institution.

Status Codes Returned by Payment Method Issuer

Code	Message	Transaction Status
00	Success	Processed
1	Refer to card issuer	Decline
3	Invalid merchant	Decline
4	Capture card	Decline
5	Do not honor	Decline
6	Customer requested stop of specific recurring payments	Decline
7	Customer requested stop of all recurring payments	Decline
8	Honor with ID	Approve
9	Unpaid items, failed negative file check	Decline
10	Duplicate check number	Decline
11	MICR error	Decline
12	Invalid transaction	Decline
13	Referral	Decline
14	Invalid card number	Decline
15	Invalid issuer	Decline
16	You are trying to refund a card that has not been previously charged in this system.	Decline
17	Amount greater than limit	Decline
18	Too many checks (over merchant or bank limit)	Decline
19	Reenter transaction	Decline
20	Issuing bank unavailable	Decline
21	Too many checks (over merchant or bank limit)	Decline
22	Try again	Decline
23	Void error	Decline
24	Invalid expiration date	Decline
25	Invalid terminal	Decline
26	Credit error	Decline
27	Fraud filter declined	Decline
28	Fraud filter for review	Decline
29	Issuing bank timeout	Decline
30	Format error	Decline
41	Lost card	Decline
43	Stolen card	Decline
51	Insufficient funds/over credit limit	Decline
52	No checking account	Decline
54	Expired card	Decline
55	Invalid PIN	Decline
57	Transaction not permitted to issuer/cardholder	Decline
58	Transaction not permitted to acquirer/terminal	Decline
61	Exceeds withdrawal limit	Decline
62	Restricted card	Decline
63	Security violation	Decline
65	Exceeds withdrawal limit count	Decline
75	Allowable number of PIN tries exceeded	Decline
76	Invalid/nonexistent "To Account" specified	Decline
77	Invalid/nonexistent "From Account" specified	Decline
78	Invalid/nonexistent account specified (general)	Decline

80	Invalid date	Decline
82	CVV data is not correct	Decline
83	Cannot verify the PIN	Decline
84	Invalid authorization life cycle	Decline
85	Not declined	Approve
93	Violation cannot complete. Have the customer call the 800 number on the back of the card to determine the issue.	Decline
94	Duplicate transaction	Decline
96	System Error	Decline
98	Approval for a lesser amount	Consult merchant provider
99	Amount Error	Decline
100	Generic Decline	Decline
101	Failed CVV Filter	Decline
199	Misc. Decline	Decline

A.5 CVV Response Codes

The following response codes are returned in the [Transaction.RequestResult] object. They are returned only if a CVV2 is passed in the transaction request and a response returned from the card issuer. These codes do not indicate whether a transaction request was successful. They indicate whether or not the CVV2 submitted matches what the issuing institution has on file.

Code	Message
M	CVV2 Match
N	CVV2 No Match
P	Not Processed
S	Merchant indicates CVV2 not present on card
U	Issuer is not certified and/or has not provided appropriate encryption keys

A.6 AVS Response Codes

The following response codes are returned in the [Transaction.RequestResult] object. They are returned by the card issuer. They do not indicate whether a transaction request was successful. They indicate the conformity of the address values passed in the request to those stored by the card issuer.

Domestic AVS Response Codes

Code	Message
A	Street address matches 5-digit and 9-digit postal code do not match
D	Exact Match
E	AVS Data is invalid, AVS is not allowed for this card type
N	Zip Code and Street Do Not Match
R	Issuer system unavailable
S	Service Not supported
U	Verification Unavailable*
W	Street Address does not match, 9 digit postal code does
X	Street Address and 9 digit postal code match
Y	Street Address and 5 digit postal code match
Z	Street Address does not match, 5 digit postal code does
0	No data provided to perform AVS check

*Returned if the U.S. bank does not support non-U.S. AVS or if the AVS in a U.S. bank is not functioning properly.

International AVS Response Codes

Code	Message
B	Address Match, postal code not verified
C	Street address and postal code do not match
G	Non-U.S. issuing bank does not support AVS
I	Address not verified
M	Exact Match
P	Zip Match

American Express Only AVS Response Codes

Code	Message
F	Name does not match, postal code matches
H	Name does not match, full AVS matches
J	Name does not match, full AVS does not match
K	Name matches, full AVS does not match
L	Name matches, postal code matches
O	Name match, Address Match, Postal Code no match
Q	Exact match
T	Name does not match, Street Address Match
V	Exact Match

Testing Environment AVS Response Codes

Code	Message
T	The AVS response code will always return: **T**

A.7 Fraud System Response Code

The following response codes are returned in the [RequestResult] object. They are generated by ProtectPay in response to the Fraud System and returned as the status of the API Request. They are unique to each Fraud System.

Threat Metrix

Status Codes Returned by Fraud Systems

Code	Message	Transaction Status
00	Success	Processed
133	Threat Metrix Score Threshold Met	Decline
134	Session ID or InputIP Address are invalid	Decline
135	Threat Metrix Account Error or Account not found	Decline

Appendix B: MerchantProfileId Settings - Supported Gateways

The following MerchantProfileId settings are supported by ProtectPay.
It is the responsibility of the merchant to obtain the appropriate values for each ProcessorField.

ProtectPay Supported Gateway and Credential Requirements

ProPay

Does not allow capture for more than initial authorization

Payment Processor	ProcessorField	Value
LegacyProPay	certStr	
	termId	
	accountNum	
	forceRecurring	

*Specific MCC codes will allow for capture more than initial authorization

Does not allow capture for more than initial authorization

Payment Processor	ProcessorField	Value
LegacyProPayCan	certStr	
	termId	
	accountNum	

*Specific MCC codes will allow for capture more than initial authorization

Does not allow capture for more than initial authorization

Payment Processor	ProcessorField	Value
ProPayGateway	AccountId	
	IdentityId	
	MerchantInfoId	

*Specific MCC codes will allow for capture more than initial authorization

Authorize.net

CVV code has not effect in their test environment

Payment Processor	ProcessorField	Value
AuthorizeNet	API_LOGIN_ID	
	API_TRANSACTION_KEY	

Braspag

Does not return very specific reasons for decline
Test environment does not mimic their production very well

Payment Processor	ProcessorField	Value
Braspag	AcquirerTranslator	
	MerchantID	

CyberSource

Requires billing email address for transaction processing
Transactions Require an Invoice Number

Payment Processor	ProcessorField	Value
CyberSource	TransactionKey	
	MerchantID	

Digital River

Not all values are required, depends on business needs

Payment Processor	ProcessorField	Value
DigitalRiver	MerchantId	
	MerchantId-HKD	
	MerchantId-MXN	
	MerchantId-USD	
	Password	
	POSID	
	TransactionChannel	
	Username	

E-Solutions

Payment Processor	ProcessorField	Value
Esolutions	ProfileId	
	ProfileKey	

Echo

CVV code has not effect in their test environment
Test environment requires phone number for processing

Payment Processor	ProcessorField	Value
Echo	echoId	
	echoPin	

Merchant E Solutions

Test environment will not allow credit transaction
Must wait between Authorize and Capture

Payment Processor	ProcessorField	Value
Esolutions	ProfileID	
	ProfileKey	

Meritus

Test environment will not settle transactions automatically
Must pass Address1 and ZipCode

Payment Processor	ProcessorField	Value
Meritus	MerchantID	
	MerchantKey	

Mtrex

Test environment will not return decline

Payment Processor	ProcessorField	Value
Mtrex	AuthenticationID	
	AuthenticationPassword	
	ConfigID	

Network Merchants (NMI)

Payment Processor	ProcessorField	Value
NetworkMerchants	API_LOGIN_ID	
	API_TRANSACTION_KEY	

Orbital/Paymentech

Test environment will not return decline
Username and PW not required if IP white-listed

Payment Processor	ProcessorField	Value
Orbital	OrbitalBin	
	OrbitalMerchantId	
	OrbitalTerminalId	
	OrbitalUsername	
	OrbitalPassword	
	OrbitalIndustryType	

Pagos Online

Payment Processor	ProcessorField	Value
PagosOnline	cuentaId	
	loginUsuarioAprobador	
	password	
	usuarioId	

PayflowPro

Does not allow capture for more than initial authorization

Payment Processor	ProcessorField	Value
PayFlowPro	Partner	
	PWD	
	USER	
	VENDOR	

PaymentXP

Test environment will not return decline
Test environment only supports JPY
Cannot perform credit transaction, must Void or Refund
Refunding unsettled transactions will void them
Does not return very specific reasons for decline

Payment Processor	ProcessorField	Value
PaymentXP	MerchantId	
	MerchantKey	

PayVision

Test environment will not return decline
Does not allow capture for more than initial authorization
Transactions require invoice number
Must submit amount for capture transaction
Must pass country for credit transaction

Payment Processor	ProcessorField	Value
PayVision	MemberId	
	MemberGuid	

Planet Payment

Payment Processor	ProcessorField	Value
PayVision	Password	
	User	

SecurePay

All transactions must include an Invoice Number

Payment Processor	ProcessorField	Value
SecurePay	MerchantId	
	Password	

VeriTrans

Test environment will allow void or refund
Test environment only supports JPY
Must submit amount for capture transaction
Does not allow capture for more than initial authorization
Does not return very specific reasons for decline

Payment Processor	ProcessorField	Value
VeriTrans	HashKey	
	SecretKey	

Web Collect

Payment Processor	ProcessorField	Value
WebCollect	MERCHANTID	

WorldPay

Payment Processor	ProcessorField	Value
WorldPay	MerchantCode	
	Password	

Appendix C: Fraud Detection

ProtectPay offers integration opportunities to various Fraud Systems to help ProtectPay Merchants from processing fraudulent credit cards and/or known fraudulent bank accounts. A special FraudDetector Object is used to pass along credentials and additional Fraud System specific information to the Fraud System Provider.

In order to use the FraudDetector Object a client must specify the Provider Name and provide the provider specific attributes as indicated in this appendix. While use of the FraudDetector object is optional, if used, specific attributes for each individual fraud system provider are required.

FraudDetector Compatible Methods

The following ProtectPay API Methods are compatible with the FraudDetector object

- 4.4.1 Authorize a PaymentMethodId
- 4.4.2 Authorize a PaymentMethodId (Recurring)
- 4.4.3 Authorize a PaymentMethodId with Encrypted Block Data
- 4.5.1 Process a PaymentMethodId
- 4.5.2 Process a PaymentMethodId (Recurring)
- 4.5.3 Process a PaymentMethodId with Encrypted Block Data
- 4.6.3 Process a Credit Transaction
- 4.9.1 ProPay SplitPay Transaction
- 4.9.2 ProPay SplitPay Transaction with Encrypted Block Data
- 4.10.1 Authorize External Transaction
- 4.10.2 Process External Transaction
- 4.10.3 Process External ProPay SplitPay Transaction
- 4.10.4 Process a Credit Card

FraudDetector Base Object:

Request Attribute	Object
FraudDetector	FraudDetectorProvider
	* Specific Attributes for Fraud System Provider

Interface: REST

The FraudDetector object is passed in the parent object for REST methods.

```
FraudDetector:{
       "FraudDetectorProviderName":"Unknown",
       /* Specific Attributes for Fraud System Provider */
}
```

Interface: SOAP

The following namespace must be added to the parent object is using SOAP-XML

```
<s:Envelope
                    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                    xmlns:con="http://propay.com/SPS/contracts"
                    xmlns:typ="http://propay.com/SPS/types"
                    xmlns:prop="http://schemas.datacontract.org/2004/07/Propay.Contracts.SPS.External"
                    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
   <s:Body>
      …
      …
   </s:Body>
</s:Envelope>
```

The Fraud Detector Object must be formatted specifically to reference this aforementioned additional namespace and must include subsequent namespaces for the FraudDetectorProviderName an additional namespace for the individual elements of the request

The FraudDetector Object itself is added to the following:

- For PaymentMethodId methods it is added to the Transaction object:

```
<prop:Transaction>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
                <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
                <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</prop:Transaction>
```

- For EncryptedBlockData methods it is added to the AuthorizeAndCapture object:

```
<con: AuthorizeAndCapture >
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
                <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
                <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: AuthorizeAndCapture >
```

- For Create HostedTransacionIdentifier method it is added to the hostedTransaction object.

```
<con:hostedTransaction>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
                <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
                <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: hostedTransaction >
```

- For ProPay SplitPay Transaction method it is added to the request object :

```
<con: request>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
                <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
                <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: request>
```

- For ProPay SplitPay Transaction with Encrypted Block Data method it is added to the ProcessSplitPayTransactionWithEncryptedTrackData object :

```
<con: ProcessSplitPayTransactionWithEncryptedTrackData>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
```

```
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
            <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
            <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: ProcessSplitPayTransactionWithEncryptedTrackData>
```

- For External Transaction methods it is added to the ExternalPaymentMethodTransaction object :

```
<con: externalPaymentMethodTransaction>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
            <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
            <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: externalPaymentMethodTransaction>
```

- For External SplitPay Transaction method it is added to the ExternalPaymentMethodSplitPayTransaction object :

```
<con: externalPaymentMethodSplitPayTransaction>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
            <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
            <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: externalPaymentMethodSplitPayTransaction>
```

- For Process a Credit Card method it is added to the ProcessCard object :

```
<con: ProcessCard>
        <typ:FraudDetector i:type="c:<!—Specific Fraud Service Provider Object Reference-->"
        xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
        xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
            <b:FraudDetectorProviderName>Unknown</b:FraudDetectorProviderName>
            <!--Specific Attributes for Fraud System Provider, note they use namespace c: -->
        </typ:FraudDetector>
</con: ProcessCard>
```

Interface: WSDL

In order to properly use the FraudDetector object by extrapolating the WSDL the specific Fraud System Provider Object must be created and set to the value of the FraudDetector.

This can be done in the following manner:

Request Attribute:	Object	Attributes
FraudDetector	**Specific Fraud Provider Object**	FraudDetectorProviderName
		Attribute1
		Attribute2
		Attribute3
		…

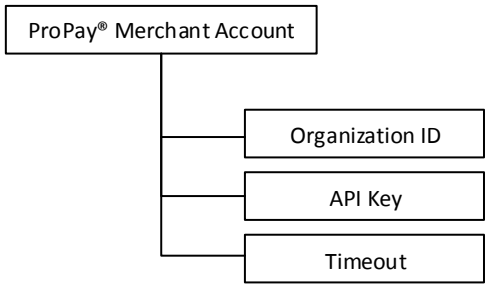
- ❖ See supported FraudDetector Providers for specific examples

Threat Metrix

Threat Metrix Account Setup

ProPay must set up a ProPay merchant account to use Threat Metrix. This cannot be done through the Application Programming Interface. The Threat Metrix credentials are tied directly to the ProtectPay BillerId and are available only to the specified ProtectPay BillerId. Please refer requests to obtain Threat Metrix account information to: riskescalation@propay.com

- If a client has access to multiple ProtectPay BillerId's they will have multiple Threat Metrix Credentials



The Organization ID is the value assigned by Threat Metrix to represent the client's ProtectPay BillerId. It must be used to create a Threat Metrix Session ID.

The API Key is the clients Threat Metrix API credentials that ProtectPay will use when consuming the Threat Metrix.

The Timeout value is a value in milliseconds the ProtectPay system will wait for a response from Threat Metrix before automatically passing the transaction along to the processor. This value is set by ProPay at 2000ms and can be adjusted by the client with a request to ProPay. If the timeout period elapses the transaction is passed to the processor which can create a case where a transaction was actually determined to be fraudulent, however the Threat Metrix API responded after the timeout period elapsed.

- Please work with the ProPay risk department to mitigate such occurrences and develop an appropriate resolution.

ProPay will supply the client a Threat Metrix username and password. The client must then sign into the Threat Metrix Portal and set up their risk profiles that are used to determine whether or not a transaction will be considered fraudulent by the client. The ProPay risk department can assist a client in determining which attributes should be set in a risk profile however it is the responsibility of the client to determine what will be considered a fraudulent transaction and what will not.

Threat Metrix Portal URI: https://portal2.threatmetrix.com
For additional information on setting up risk profiles please see: https://kb.threatmetrix.com/index.php?View=login&Msg=_index

Threat Metrix SessionId Creation

Prior to sending a transaction request to the ProtectPay API the merchant must create and send to Threat Metrix a unique SessionId. Threat Metrix hosts a download of an invisible iFrame that must be placed on the merchant's website prior to the checkout page. ProPay recommends the use of an order confirmation page to accomplish this prior to navigation to the final checkout page.

The Threat Metrix iFrame requires that the appropriate organization ID be sent. The Threat Metrix iFrame gathers information from the payer's browser and associates with the SessionId that must be passed to Threat Metrix. It is important that this SessionId is persisted in the browser session to the final checkout page as it must be passed to ProPay in the API call.

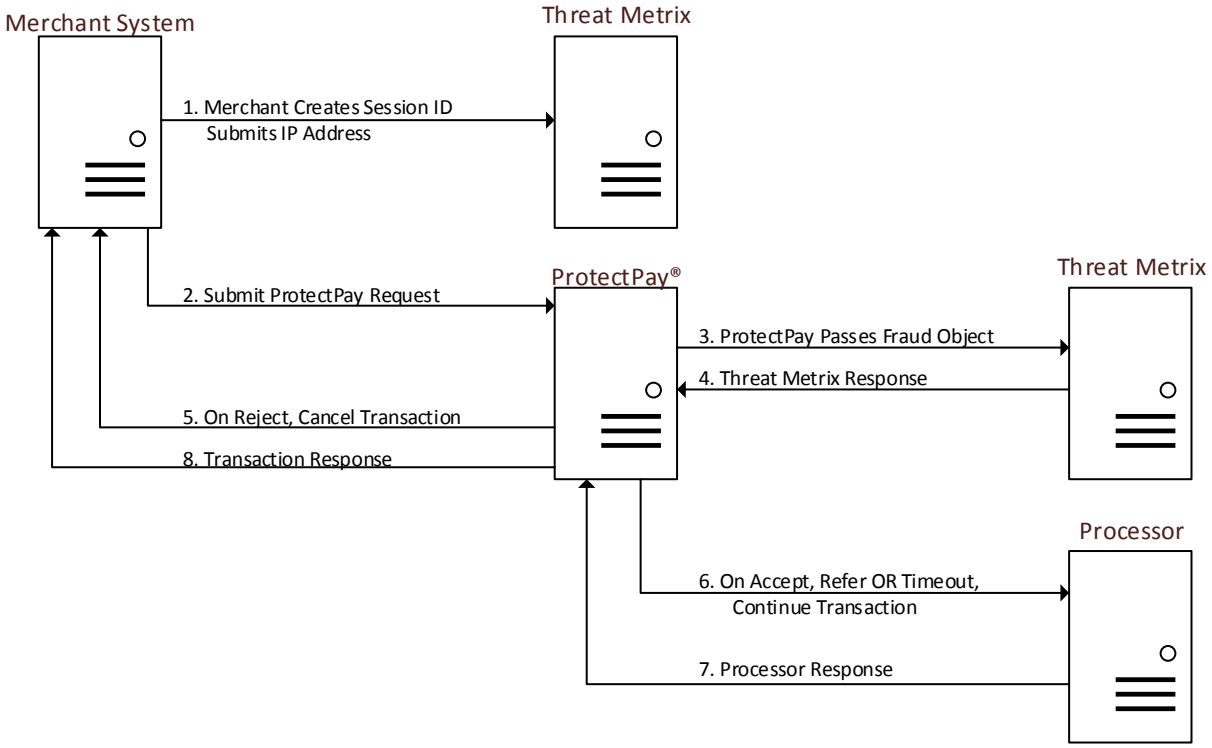
Threat Metrix Processing flow

1. Merchant System Creates Threat Metrix Session ID and Submits Input IP Address of payers web browser.
2. Merchant System Submits ProPay API Request including Fraud Object.
    a. See Object attributes below
    b. 60 second ProtectPay timeout timer begins
3. ProPay Submits Fraud Object to Threat Metrix including Session ID, Input IP Address and Filter Requirements.
4. Threat Metrix responds with score and following messages
    a. Accept
    b. Refer
    c. Reject
    d. Error
5. On Reject or Error the transaction is cancelled and reported back to the Merchant with appropriate response code.
    a. See Appendix A.7 Fraud System Response Codes: Threat Metrix.
    b. The Actual Score is not returned. Please log into the Threat Metrix Portal to view scores.

6. On Accept, Refer or at timeout the Transaction is passed to the Processor.
    a. The Threat Metrix timeout period is part of the ProtectPay 60 second timeout and does not extend it.
7. The Processor responds to the transaction request.
8. ProtectPay responds to the merchant with the transaction response.

- Both the SessionId and IP Address must be passed to else the Threat Metrix process is ignored

Threat Metrix Process flow diagram



Threat Metrix Specific Attributes

Attribute	Type	Max	Required	Notes
FraudDetectorProvider	String		Required	Set to: ThreatMetrix
SessionId	String		Required	Created by merchant and sent to Threat Metrix prior to transaction
InputIpAddress	String		Required	Sent by merchant to Threat Metrix prior to transaction
ShippingAddress1	String		Optional	
ShippingAddress2	String		Optional	
ShippingCity	String		Optional	
ShippingState	String		Optional	
ShippingZip	String		Optional	
ShippingCountry	String		Optional	
CustomAttribute1	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute2	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute3	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute4	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute5	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute6	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute7	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute8	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute9	String		Optional	Must exist as part of the Organization Id prior to being passed
CustomAttribute10	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute1	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute2	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute3	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute4	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute5	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute6	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute7	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute8	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute9	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute10	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute11	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute12	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute13	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute14	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute15	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute16	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute17	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute18	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute19	String		Optional	Must exist as part of the Organization Id prior to being passed
ConditionalAttribute20	String		Optional	Must exist as part of the Organization Id prior to being passed

Interface: REST


```
FraudDetector:{
        "FraudDetectorProviderName":"ThreatMetrix",
        "SessionId":"08a3958c-f2f5-43ad-b171-9de35633ff68",
        "InputIpAddress":"8.8.8.8",
        "ShippingAddress1":"",
        "ShippingAddress2":"",
        "ShippingCity":"",
        "ShippingState":"",
        "ShippingZip":"",
        "ShippingCountry":"",
        "ConditionalAttribute1":"",
        "ConditionalAttribute2":"",
        "ConditionalAttribute3":"",
        "ConditionalAttribute4":"",
        "ConditionalAttribute5":"",
        "ConditionalAttribute6":"",
        "ConditionalAttribute7":"",
        "ConditionalAttribute8":"",
        "ConditionalAttribute9":"",
        "ConditionalAttribute10":"",
        "ConditionalAttribute11":"",
        "ConditionalAttribute12":"",
        "ConditionalAttribute13":"",
        "ConditionalAttribute14":"",
        "ConditionalAttribute15":"",
        "ConditionalAttribute16":"",
        "ConditionalAttribute17":"",
        "ConditionalAttribute18":"",
        "ConditionalAttribute19":"",
        "ConditionalAttribute20":"",
        "CustomAttribute1":"",
        "CustomAttribute2":"",
        "CustomAttribute3":"",
        "CustomAttribute4":"",
        "CustomAttribute5":"",
        "CustomAttribute6":"",
        "CustomAttribute7":"",
        "CustomAttribute8":"",
        "CustomAttribute9":"",
        "CustomAttribute10":""
}
```

Interface: SOAP

```
<typ:FraudDetector i:type="c:ThreatMetrixFraudDetection"
xmlns:b="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection"
xmlns:c="http://schemas.datacontract.org/2004/07/Propay.Contracts.FraudDetection.Providers">
        <b:FraudDetectorProviderName>ThreatMetrix</b:FraudDetectorProviderName>
        <c:SessionId>0b0adfe1-f7fe-4c0a-9478-941cbabc6b18</c:SessionId>
        <c:ConditionalAttribute1></c:ConditionalAttribute1>
        <c:ConditionalAttribute10></c:ConditionalAttribute10>
        <c:ConditionalAttribute11></c:ConditionalAttribute11>
        <c:ConditionalAttribute12></c:ConditionalAttribute12>
        <c:ConditionalAttribute13></c:ConditionalAttribute13>
        <c:ConditionalAttribute14></c:ConditionalAttribute14>
        <c:ConditionalAttribute15></c:ConditionalAttribute15>
        <c:ConditionalAttribute16></c:ConditionalAttribute16>
        <c:ConditionalAttribute17></c:ConditionalAttribute17>
        <c:ConditionalAttribute18></c:ConditionalAttribute18>
        <c:ConditionalAttribute19></c:ConditionalAttribute19>
        <c:ConditionalAttribute2></c:ConditionalAttribute2>
        <c:ConditionalAttribute20></c:ConditionalAttribute20>
        <c:ConditionalAttribute3></c:ConditionalAttribute3>
        <c:ConditionalAttribute4></c:ConditionalAttribute4>
        <c:ConditionalAttribute5></c:ConditionalAttribute5>
        <c:ConditionalAttribute6></c:ConditionalAttribute6>
        <c:ConditionalAttribute7></c:ConditionalAttribute7>
        <c:ConditionalAttribute8></c:ConditionalAttribute8>
        <c:ConditionalAttribute9></c:ConditionalAttribute9>
        <c:CustomAttribute1></c:CustomAttribute1>
        <c:CustomAttribute10></c:CustomAttribute10>
        <c:CustomAttribute2></c:CustomAttribute2>
        <c:CustomAttribute3></c:CustomAttribute3>
        <c:CustomAttribute4></c:CustomAttribute4>
        <c:CustomAttribute5></c:CustomAttribute5>
        <c:CustomAttribute6></c:CustomAttribute6>
        <c:CustomAttribute7></c:CustomAttribute7>
        <c:CustomAttribute8></c:CustomAttribute8>
        <c:CustomAttribute9></c:CustomAttribute9>
        <c:InputIpAddress>8.8.8.8</c:InputIpAddress>
        <c:SessionId>08a3958c-f2f5-43ad-b171-9de35633ff68</c:SessionId>
        <c:ShippingAddress1></c:ShippingAddress1>
        <c:ShippingAddress2></c:ShippingAddress2>
        <c:ShippingCity></c:ShippingCity>
        <c:ShippingCountry></c:ShippingCountry>
        <c:ShippingState></c:ShippingState>
        <c:ShippingZip></c:ShippingZip>
</a:FraudDetector>
```

Interface: WSDL

FraudDetectorProviderName: Threat Metrix

Request Attribute	Object	Attributes
FraudDetector	**ThreatMetrixFraudDetection**	FraudDetectorProviderName
		SessionId
		InputIpAddress
		ShippingAddress1
		ShippingAddress2
		ShippingCity
		ShippingState
		ShippingZip
		ShippingCountry
		ConditionalAttribute1
		ConditionalAttribute2
		ConditionalAttribute3
		ConditionalAttribute4
		ConditionalAttribute5
		ConditionalAttribute6
		ConditionalAttribute7
		ConditionalAttribute8
		ConditionalAttribute9
		ConditionalAttribute10
		ConditionalAttribute12
		ConditionalAttribute13
		ConditionalAttribute14
		ConditionalAttribute15
		ConditionalAttribute16
		ConditionalAttribute17
		ConditionalAttribute18
		ConditionalAttribute19
		ConditionalAttribute20
		CustomAttribute1
		CustomAttribute2
		CustomAttribute3
		CustomAttribute4
		CustomAttribute5
		CustomAttribute6
		CustomAttribute7
		CustomAttribute8
		CustomAttribute9
		CustomAttribute10

Appendix D: ProtectPay Supported Swipe Devices

ProPay approved swipe devices encrypt credit card track data at the head as the card is swiped. The encrypted data is then transmitted to the connected device as an encrypted block. Elements of the encrypted block can be submitted to ProtectPay.

Supported ProtectPay Swipe Devices

Make	Model	Part Number
Dynamag	MagTek Dynamag	21073075
FLASH Card Reader 1.0	MagTek MagneSafe m20	21073034 (Rev–F)
FLASH Card Reader 2.0	MagTek flash	21073081 (Rev–C)
JAK 1.0	ID Tech Unimag	ID-80110001-001 (Rev–H)
JAK 1.1	ID Tech Unimag Pro	ID-80110004-001 (Rev–C)
JAK 2.0	ID Tech Shuttle	ID-80110010-010 (Rev–A)
JAK 3.0	Magtek aDynamo	21073111
JAK 4.0	Roam	G5X

D.1 Supported ProtectPay API Methods

The following ProtectPay API methods accept encrypted track data:

- 4.3.2 Create a PaymentMethodId with Encrypted Block Data
- 4.4.3 Authorize a Payment Method with Encrypted Block Data
- 4.5.3 Process a Payment Method with Encrypted Block Data
- 4.8.2 ProPay SplitPay Transaction with Encrypted Block Data

D.2 Swipe Device Enumeration Map

The following lists the enumeration list that is used when identifying a swipe device in an API method. Either the name of the device can be passed or the numerical value.

Device Type Enumerative List

Value	Device Type
1	MagTekM20
2	MagTekFlash
3	IdTechUniMag
4	Manual
5	MagTekADynamo
6	MagTekDynamag
7	RoamData

D.3 Software Development Kit

ProPay offers a .NET Software Development Kit for the Dynamag to assist developers in incorporating a swipe device into their developed or developing software solution. Please request additional information from techincalsupport@propay.com

Appendix E: EnsureBill for ProtectPay

EnsureBill is a service by which card numbers and expiration dates can be updated as new information is available from the issuing banks. Clients must request enrollment in EnsureBill through their relationship manager, and should specify whether they want all active cards stored in ProtectPay to be updated, or only cards that have been marked as protected (cards used for recurring billing, for example).

At enrollment, all previously-stored cards can be enrolled for updates.

Once enrolled, clients will receive a report via email or sFTP indicating which payment methods have been updated and the new details of those updates (new obfuscated card number or expiration date). The client system may need to be enhanced to read in the response files so the client system can be updated and reflect the most current details of the payment methods as they are updated.

Appendix F: ProtectPay Data Import

ProtectPay hosts an interface to import existing sensitive payment method information in a secure manner. A merchant can generate a formatted XML file and upload it to ProPay's sFTP server to be imported.

F.1 Supported ProtectPay API Methods

The following ProtectPay API methods are exposed for data importing:

- Create a PayerId
  - Create PaymentMethodId
- Edit a PayerId
  - Create PaymentMethodId
  - Edit PaymentMethodId
  - Delete PaymentMethodId
- Delete a Payer
  - Will also delete all PaymentMethodIds for the PayerId

F.2 XML File Creation

In order to upload data, the client system must generate a correctly formatted XML file that is submitted to ProtectPay for importing. The data import interface does not check the validity of card data, including expiration dates when importing information. It will import what is requested to be imported. The client must check the validity of the card data prior to submitting it for import if it is intended to be used for processing.

Multiple requests for services can be combined into one XML transmission

The API allows for multiple requests to be incorporated into a single file upload. ProtectPay replies to each nested request by returning the result code of the request. In the event that a single request fails, the additional requests will attempt to be processed.

*See section F.4 for additional information about XML file formatting per method request.
*See section A for additional information about responses returned by the interface.

F.3 Uploading and Processing the XML File

ProtectPay uses sFTP to receive sensitive payment method data and can return responses via email or sFTP. Once a file has been prepared, a login for the ProtectPay secure transfer website, https://xfer.propay.com, can be requested from a ProPay sales representative and/or account manager.

- The client will need to supply the IP address(es) of the server and/or computer from which the file(s) will be transmitted.

ProPay will respond with the login credentials and the URL for submitting secure files.

*In order to upload the file via the web, the client Browser MUST support ActiveX.
Client will upload the file to ProPay's sFTP server and email ***technicalsupport@propay.com*** to notify ProPay there is a new file that needs to be processed.

Receiving Response Files

Once the file is processed, ProtectPay will produce a response XML file. This file will be placed back on the secure transfer site where it can be downloaded and read into the client system. A sample response file is available upon request.

F.4 Data Import Methods Defined

Create a PayerId

This method will create a new ProtectPay PayerId and a PaymentMethodId. ProtectPay will respond to this request by mirroring the data back to the sender. If also creating a PaymentMethodId, this method does not check the validity of card data including expiration dates when importing information. A user must check the validity of the card data prior to submitting it for import.

Request values defined

Request Attribute	Notes
BatchCommandRequest	
BatchCommandRequest{UniqueId}	This value is set by the client and is echoed back for client system linking.
BatchCommandRequest{AuthenticationToken}	Used to access the API.
System	
System{Id}	Set to "SPS"
System{BillerId}	Used to identify the correct collection of PayerIds and PaymentMethodIds.
Command	
Command{UniqueId}	Set to "1"
Command{Type}	Set to "ADDPAYER"
Payer	
Payer{Name}	Used to identify a payer.
PaymentMethods[]	Collection of payment methods
PaymentMethods[].PaymentMethod	
PaymentMethods[].PaymentMethod{Action}	Used to indicate the action to perform on the PaymentMethodId ADD EDIT DELETE
PaymentMethods[].PaymentMethod{Priority}	Used to explicitly set an order for the ProcessPayment transaction.
PaymentMethods[].PaymentMethod{Type}	Used to tell ProtectPay what type of data is being submitted. Valid values are: ▪ Visa ▪ MasterCard ▪ AMEX ▪ Discover ▪ DinersClub ▪ JCB ▪ ProPayToProPay ▪ Checking ▪ Savings
PaymentMethods[].PaymentMethod.AccountNumber	Used to identify a payer.
PaymentMethods[].PaymentMethod.ExpirationDate	The expiration date for a payment method. For a credit card these are submitted as 4 digit numeric values MMYY. Expiration dates are optional but if the system needs an expiration date in order to process, you need to either add it here or supply it as an optional payment method override when performing a transaction.
PaymentMethods[].PaymentMethod.BillingAddress1	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingAddress2	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingCity	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingState	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingZipCode	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingCountry	ISO 3166 standard 3 character country codes. Current allowed values are: USA CAN

Response values defined

Response Attribute	Notes

ResultValue	The ProtectPay API Method Response Value.
ResultCode	The ProtectPay API Method Response Code. See Appendix A for possible returned values.
ResultMessage	The ProtectPay API Method Response Message. See Appendix A for possible returned messages.
ExternalAccountID	This is the ProtectPay ID for the Payer Created and belongs to the BillerID that created it. *This is referenced in other methods as 'PayerAccountID' or 'PayerID'.
PaymentMethodID	This is the ProtectPay ID for the Payment Method, also called a Token. The Payment Method Created Belongs to the PayerId for which it was created.

Example of XML file request

```
<?xml version="1.0" ?>
<BatchCommandRequest UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66" AuthenticationToken="68FA7603-05B8-4725-89A0-689154067CA2">
 <System Id="SPS" BillerExternalId="564738291346789">
  <Command UniqueId="1" Type="ADDPAYER">
   <Payer Name="Flint King">
    <PaymentMethods>
     <PaymentMethod Priority="1" Type="VISA">
      <AccountName>Flint King</AccountName>
      <AccountNumber>4747474747474747</AccountNumber>
      <ExpirationDate>0110</ExpirationDate>
      <BillingAddress1>1234 Anystreet Rd.</BillingAddress1>
      <BillingAddress2 />
      <BillingCity>Sandy</BillingCity>
      <BillingState>UT</BillingState>
      <BillingZipcode>84092</BillingZipcode>
      <BillingCountry>USA</BillingCountry>
     </PaymentMethod>
    </PaymentMethods>
   </Payer>
  </Command>
 </System>
</BatchCommandRequest>
```

Example of XML data returned from ProtectPay for the 'Add Payer' function:

```
<?xml version="1.0" ?>
<BatchCommandResponse UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66">
 <System Id="SPS" BillerExternalId="564738291346789">
  <Command UniqueId="1" Type="ADDPAYER">
   <Result>SUCCESS</Result>
   <ResultCode>00</ResultCode>
   <ResultMessage />
   <Payer Name="Flint King" ExternalId="2345678998765432">
    <PaymentMethods>
     <PaymentMethod Priority="1" Type="VISA" PaymentMethodId="3E6FF1BE-3620-4ee5-BFF0-876A9A429EA5">
      <Result>SUCCESS</Result>
      <ResultCode>00</ResultCode>
      <ResultMessage />
     </PaymentMethod>
    </PaymentMethods>
   </Payer>
  </Command>
 </System>
</BatchCommandResponse>
```

Edit a PayerId

This method will edit a PayerId as well as create additional PaymentMethodId, edit PaymentMethodId or delete PaymentMethodId from a specific PayerId. ProtectPay will respond to this request by mirroring the data back to the sender. This method does not check the validity of card data including expiration dates when importing information. A user must check the validity of the card data prior to submitting it for import.

Request values defined

Request Attribute	Notes
BatchCommandRequest	
BatchCommandRequest{UniqueId}	This value is set by the client and is echoed back for client system linking.
BatchCommandRequest{AuthenticationToken}	Used to access the API.
System	
System{Id}	Set to "SPS".
System{BillerId}	Used to identify the correct collection of PayerId's and PaymentMethodId's.
Command	
Command{UniqueId}	Set to "2".
Command{Type}	Set to "EDITPAYER".
Payer	
Payer{Name}	Used to identify a payer.
PaymentMethods[]	
PaymentMethods[].PaymentMethod	
PaymentMethods[].PaymentMethod{Action}	Used to indicate the action to perform on the PaymentMethodId. ADD EDIT DELETE
PaymentMethods[].PaymentMethod{Priority}	Used to explicitly set an order for the ProcessPayment transaction.
PaymentMethods[].PaymentMethod{Type}	Used to tell ProtectPay what type of data is being submitted. Valid values are: Visa MasterCard AMEX Discover DinersClub JCB ProPayToProPay Checking Savings
PaymentMethods[].PaymentMethod.AccountNumber	Used to identify a payer.
PaymentMethods[].PaymentMethod.ExpirationDate	The expiration date for a payment method. For a credit card these are submitted as 4 digit numeric values MMYY. Expiration dates are optional but if the system needs an expiration date in order to process, you need to either add it here or supply it as an optional payment method override when performing a transaction.
PaymentMethods[].PaymentMethod.BillingAddress1	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingAddress2	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingCity	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingState	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingZipCode	The address on the account for a payment method.
PaymentMethods[].PaymentMethod.BillingCountry	ISO 3166 standard 3 character country codes. Current allowed values are: USA CAN

Response values defined

Response Attribute	Notes
ResultValue	The ProtectPay API Method Response Value.
ResultCode	The ProtectPay API Method Response Code. See Appendix A for possible returned values.
ResultMessage	The ProtectPay API Method Response Message. See Appendix A for possible returned messages.
ExternalAccountID	This is the ProtectPay ID for the Payer Created and belongs to the BillerID that created it. *This is referenced in other methods as 'PayerAccountID' or 'PayerID'.
PaymentMethodID	This is the ProtectPay ID for the Payment Method, also called a Token. The Payment Method Created Belongs to the PayerId for which it was created.

Example of XML file request

```
<?xml version="1.0" ?>
<BatchCommandRequest UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66" AuthenticationToken="68FA7603-05B8-4725-89A0-689154067CA2">
 <System Id="SPS" BillerExternalId="564738291346789">
  <Command UniqueId="1" Type="EDITPAYER">
   <Payer ExternalId="2345678998765432">
    <PaymentMethods>
     <PaymentMethod Action="ADD" Priority="1" Type="VISA">
      <AccountName>Flint King</AccountName>
      <AccountNumber>4747474747474747</AccountNumber>
      <ExpirationDate>0110</ExpirationDate>
      <BillingAddress1>1234 Anystreet Rd</BillingAddress1>
      <BillingAddress2 />
      <BillingCity>Sandy</BillingCity>
      <BillingState>UT</BillingState>
      <BillingZipcode>84092</BillingZipcode>
      <BillingCountry>USA</BillingCountry>
     </PaymentMethod>
    </PaymentMethods>
   </Payer>
  </Command>
  <Command UniqueId="1" Type="EDITPAYER">
   <Payer ExternalId="9345677898767652">
    <PaymentMethods>
     <PaymentMethod Action="DELETE" PaymentMethodId="3dabb760-10bb-11dd-bd0b-0800200c9a67"/>
    </PaymentMethods>
   </Payer>
  </Command>
 </System>
</BatchCommandRequest>
```

Example of XML data returned from ProtectPay for the 'Add Payer' function:

```
<?xml version="1.0" ?>
<BatchCommandResponse UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66">
 <System Id="SPS" BillerExternalId="564738291346789">
  <Command UniqueId="2" Type="EDITPAYER">
   <Payer ExternalId="2345678998765432">
    <PaymentMethods>
     <PaymentMethod Action="ADD" Priority="1" Type="VISA" PaymentMethodId="2ED66911-EFD9-4f3d-8665-A0052FB4320A">
      <Result>SUCCESS</Result>
      <ResultCode>00</ResultCode>
      <ResultMessage />
     </PaymentMethod>
     <PaymentMethod Action="EDIT" PaymentMethodId="2dabb760-10bb-11dd-bd0b-0800200c9a66">
      <Result>SUCCESS</Result>
      <ResultCode>00</ResultCode>
```

```
        <ResultMessage />
       </PaymentMethod>
       <PaymentMethod Action="DELETE" PaymentMethodId="3dabb760-10bb-11dd-bd0b-0800200c9a67">
        <Result>FAILED</Result>
        <ResultCode>22</ResultCode>
        <ResultMessage>Invalid Payment Method Id</ResultMessage>
       </PaymentMethod>
      </PaymentMethods>
     </Payer>
    </Command>
   </System>
</BatchCommandResponse>
```

Delete a PayerId

This method will delete a PayerId and all associated PaymentMethodId. A PayerId that is deleted is no longer available for use by the owning BillerId. A PaymentMethodId that is deleted is no longer available for use by the owning PayerId.

Request values defined

Request Attribute	Notes
BatchCommandRequest	
BatchCommandRequest{UniqueId}	This value is set by the client and is echoed back for client system linking.
BatchCommandRequest{AuthenticationToken}	Used to access the API.
System	
System{Id}	Set to “SPS”.
System{BillerId}	Used to identify the correct collection of PayerId’s and PaymentMethodId’s.
Command	
Command{UniqueId}	Set to “3”.
Command{Type}	Set to “DELETEPAYER”.
Payer	

Response values defined

Response Attribute	Notes
ResultValue	The ProtectPay API Method Response Value.
ResultCode	The ProtectPay API Method Response Code. See Appendix A for possible returned values.
ResultMessage	The ProtectPay API Method Response Message. See Appendix A for possible returned messages.

Example of XML file request

```
<?xml version="1.0" ?>
<BatchCommandRequest UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66" AuthenticationToken="68FA7603-05B8-4725-89A0-689154067CA2">
 <System Id=“SPS” BillerExternalId="564738291346789">
  <Command UniqueId="3" Type="DELETEPAYER">
   <Payer ExternalId="2345678998765432" />
  </Command>
 </System>
</BatchCommandRequest >
```

Example of XML data returned from ProtectPay for “Delete Payer” function:

```
<?xml version="1.0" ?>
<BatchCommandResponse UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66">
 <System Id=“SPS” BillerExternalId="564738291346789">
  <Command UniqueId="3" Type="DELETEPAYER">
   <Result>SUCCESS</Result>
   <ResultCode>00</ResultCode>
   <ResultMessage />
   <Payer ExternalId="2345678998765432" />
  </Command>
 </System>
</BatchCommandResponse>
```

Multiple request types

Example of XML file request

```
<?xml version="1.0" ?>
<BatchCommandRequest UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66" AuthenticationToken="68FA7603-05B8-4725-89A0-689154067CA2">
 <System Id="SPS" BillerExternalId="564738291346789">
  <Command UniqueId="1" Type="ADDPAYER">
   <Payer Name="Flint King">
    <PaymentMethods>
     <PaymentMethod Priority="1" Type="VISA">
      <AccountName>Flint King</AccountName>
      <AccountNumber>4747474747474747</AccountNumber>
      <ExpirationDate>0110</ExpirationDate>
      <BillingAddress1>1234 Anystreet Rd.</BillingAddress1>
      <BillingAddress2 />
      <BillingCity>Sandy</BillingCity>
      <BillingState>UT</BillingState>
      <BillingZipcode>84092</BillingZipcode>
      <BillingCountry>USA</BillingCountry>
     </PaymentMethod>
    </PaymentMethods>
   </Payer>
  </Command>
  <Command UniqueId="2" Type="EDITPAYER">
   <Payer ExternalId="2345678998765432">
   <PaymentMethods>
    <PaymentMethod Action="ADD" Priority="1" Type="VISA">
     <AccountName>Flint King</AccountName>
     <AccountNumber>4747474747474747</AccountNumber>
     <ExpirationDate>0110</ExpirationDate>
     <BillingAddress1>1234 Anystreet Rd.</BillingAddress1>
     <BillingAddress2 />
     <BillingCity>Sandy</BillingCity>
     <BillingState>UT</BillingState>
     <BillingZipcode>84092</BillingZipcode>
     <BillingCountry>USA</BillingCountry>
     </PaymentMethod>
    <PaymentMethod Action="EDIT" PaymentMethodId="2dabb760-10bb-11dd-bd0b-0800200c9a66">
     <AccountName>Flint King</AccountName>
     <ExpirationDate>1210</ExpirationDate>
     <BillingAddress1>1234 Anystreet Rd.</BillingAddress1>
     <BillingAddress2 />
     <BillingCity>Sandy</BillingCity>
     <BillingState>UT</BillingState>
     <BillingZipcode>84092</BillingZipcode>
     <BillingCountry>USA</BillingCountry>
    </PaymentMethod>
    <PaymentMethod Action="DELETE" PaymentMethodId="3dabb760-10bb-11dd-bd0b-0800200c9a67" />
    </PaymentMethods>
   </Payer>
  </Command>
  <Command UniqueId="3" Type="DELETEPAYER">
   <Payer ExternalId="2345678998765432" />
  </Command>
 </System>
</BatchCommandRequest>
```

Example of XML data returned from ProtectPay for multiple transactions:

```
<?xml version="1.0" ?>
<BatchCommandResponse UniqueId="9951cc70-10b6-11dd-bd0b-0800200c9a66">
 <Result>Success</Result>
 <ResultCode>00</ResultCode>
 <ResultMessage />
 <System Id="SPS" BillerExternalId="564738291346789">
  <Result>Success</Result>
  <ResultCode>00</ResultCode>
  <ResultMessage />
  <Command UniqueId="1" Type="ADDPAYER">
   <Result>SUCCESS</Result>
   <ResultCode>00</ResultCode>
   <ResultMessage />
   <Payer Name="Flint King" ExternalId="2345678998765432">
    <PaymentMethods>
     <PaymentMethod Priority="1" Type="VISA" PaymentMethodId="3E6FF1BE-3620-4ee5-BFF0-876A9A429EA5">
      <Result>SUCCESS</Result>
      <ResultCode>00</ResultCode>
      <ResultMessage />
     </PaymentMethod>
    </PaymentMethods>
   </Payer>
  </Command>
  <Command UniqueId="2" Type="EDITPAYER">
   <Payer ExternalId="2345678998765432">
   <PaymentMethods>
    <PaymentMethod Action="ADD" Priority="1" Type="VISA" PaymentMethodId="2ED66911-EFD9-4f3d-8665-A0052FB4320A">
     <Result>SUCCESS</Result>
     <ResultCode>00</ResultCode>
     <ResultMessage />
    </PaymentMethod>
    <PaymentMethod Action="EDIT" PaymentMethodId="2dabb760-10bb-11dd-bd0b-0800200c9a66">
     <Result>SUCCESS</Result>
     <ResultCode>00</ResultCode>
     <ResultMessage />
    </PaymentMethod>
    <PaymentMethod Action="DELETE" PaymentMethodId="3dabb760-10bb-11dd-bd0b-0800200c9a67">
     <Result>FAILED</Result>
     <ResultCode>22</ResultCode>
     <ResultMessage>Invalid Payment Method Id</ResultMessage>
    </PaymentMethod>
   </PaymentMethods>
   </Payer>
  </Command>
  <Command UniqueId="3" Type="DELETEPAYER">
   <Result>SUCCESS</Result>
   <ResultCode>00</ResultCode>
   <ResultMessage />
   <Payer ExternalId="2345678998765432" />
  </Command>
 </System>
</BatchCommandResponse>
```