Merchant Hosted – PxPost Integration Guide

Version 1.0

paymentexpress®

CONTENTS

OVERVIEW

PxPost is designed to handle transactions using a HTTPS POST Request. The XML is generated at the client site and sent to https://sec.paymentexpress.com/pxpost.aspx.

There is no Payment Express software needed on the client side, which makes it platform and language independent. This allows for greater flexibility & interoperability.

KEY FEATURES AND SPECIFICATIONS

- No Payment Express Software needed
- Multiple Account Selection
  Transactions can be redirected to different merchant accounts depending on the credentials (Username/Password) that is specified with each transaction.
- Risk Management Rules can be enabled
- Optional reference fields for reconciliation and holding of information that will appear on transaction reports.
- Multi-Currency Support
- SSL supported

TRANSACTION INPUT - XML

Transaction variables are POSTed to the PxPost.aspx in XML form as a POST request to the following endpoint: https://sec.paymentexpress.com/pxpost.aspx.

Sample XML Transaction Input

```
<Txn>
 <PostUsername>TestUsername</PostUsername>
 <PostPassword>TestPassword</PostPassword>
 <CardHolderName>A Anderson</CardHolderName>
 <CardNumber>4111111111111111</CardNumber>
 <Amount>1.23</Amount>
 <DateExpiry>1010</DateExpiry>
 <Cvc2>3456</Cvc2>
 <Cvc2Presence>1</Cvc2Presence>
 <InputCurrency>NZD</InputCurrency>
 <TxnType>Purchase</TxnType>
 <TxnId>inv1278</TxnId>
 <MerchantReference>Test Transaction</MerchantReference>
</Txn>
```

INPUT ELEMENTS SPECIFICATIONS

Parameter	Required	Description
Amount	Yes	Amount of Transaction (dddddd.cc)
CardHolderName	No	Card holder's name
CardNumber	No	Card Number
BillingId	No	Needs to be generated to add a card for recurring billing and sent again when rebilling transactions.
Cvc2	No	Card Verification number. This number is found on the back of a credit card in the signature panel - it is different from the embossed card number and provides an additional safety check.
Cvc2Presence	No	This field is used as a CVC presence indicator. It is provided by the card acceptor to indicate availability of the CVC value on the card. Values are 0, 1, 2 & 9.
DateExpiry	No	Expiry Date on Card
DpsBillingId	No	The BillingId generated by Payment Express when adding a card for recurring billing. Needed for rebilling transactions when you don't use your own BillingId.
DpsTxnRef	No	Output from an original transaction request. Is a required field to do second stage transactions like Refund and Complete.
EnableAddBillCard	No	Needed for recurring billing transactions when adding a card to thePayment Express system. Set element to 1 for true and 0 for false
InputCurrency	Yes	You will need to specify a three-character currency code here.

MerchantReference	No	Optional Reference to Appear on Transaction Reports Max 64 Characters
PostUsername	Yes	Username of Account (Supplied by Payment Express)
PostPassword	Yes	Password of Account (Supplied by Payment Express)
ReceiptEmail	No	Store a card holder or customer's email address on the transaction details. It will be returned in the transaction response. The API account can be configured by PX Support to send out a transaction confirmation email automatically.
RecurringMode	No	Set specific string value required to specify the card storage reason when tokenising and rebilling a card with a token. The string value depends on the business case when tokenising and rebilling a card. Please see Token Billing section for possible string values and their usage details.
TxnType	Yes	'Purchase', 'Auth', 'Complete', 'Refund', 'Validate'
TxnData1	No	Optional Free Text
TxnData2	No	Optional Free Text
TxnData3	No	Optional Free Text
TxnId	No	Used for checking the status of a transaction
EnableAvsData	No	Address Verification System property. Values are 1 (Enable Verification), 0 (Disable Verification).
AvsAction	No	Address Verification System property. Values are 0,1 & 2.
		0 - Don't check AVS details with acquirer, but pass them through to Payment Express only.
		1 - Attempt AVS check. If the acquirer doesn't support AVS or is unavailable, then transaction will proceed as normal. If AVS is supported it will check the transaction and give the result.
		2 - The transactions needs to be checked by AVS, even if isn't available, otherwise the transaction will be blocked.
AvsPostCode	No	Address Verification System property. Post Code that is listed on the customer's bank statement
AvsStreetAddress	No	Address Verification System property. Address that is listed on the customer's bank statement.
DateStart	No	The Issue date of the customer's credit card, if Issuer requires this field to be present.
IssueNumber	No	The Issue Number of your credit card if Issuer requires this field to be present.
Track2	No	Extracted from Track2 of credit card.

XML TRANSACTION OUTPUT

The Response to the POST is in XML form. The status of a transaction is indicated by the Authorized element (0 = Declined, 1 = Accepted).

Sample XML Response/Output

In the following example, a transaction for $1.23 is requested for cardholder A Anderson, card number is 4111111111111111, Expiry date is Oct 2010, and Merchant Reference for the transaction is "Test Transaction".

```
<Txn>
 <Transaction success="1" reco="00" responseText="APPROVED"
pxTxn="true">
   <Authorized>1</Authorized>
   <ReCo>00</ReCo>
   <RxDate>20090610225432</RxDate>
   <RxDateLocal>20090611105432</RxDateLocal>
   <LocalTimeZone>NZT</LocalTimeZone>
   <MerchantReference>Test Transaction</MerchantReference>
   <CardName>Visa</CardName>
   <Retry>0</Retry>
   <StatusRequired>0</StatusRequired>
   <AuthCode>105430</AuthCode>
   <AmountBalance>0.00</AmountBalance>
   <Amount>1.23</Amount>
   <CurrencyId>840</CurrencyId>
   <InputCurrencyId>840</InputCurrencyId>
   <InputCurrencyName>USD</InputCurrencyName>
   <CurrencyRate>1.00</CurrencyRate>
   <CurrencyName>USD</CurrencyName>
   <CardHolderName>A ANDERSON</CardHolderName>
   <DateSettlement>20090611</DateSettlement>
   <TxnType>Purchase</TxnType>
   <CardNumber>411111........11</CardNumber>
   <TxnMac>BD43E619</TxnMac>
   <DateExpiry>1010</DateExpiry>
   <ProductId></ProductId>
   <AcquirerDate>20090611</AcquirerDate>
   <AcquirerTime>105430</AcquirerTime>
   <AcquirerId>9001</AcquirerId>
   <Acquirer>Undefined</Acquirer>
   <AcquirerReCo></AcquirerReCo>
   <AcquirerResponseText></AcquirerResponseText>
   <TestMode>0</TestMode>
   <CardId>2</CardId>
   <CardHolderResponseText>APPROVED</CardHolderResponseText>
   <CardHolderHelpText>The Transaction was approved</CardHolderHelpText>
   <CardHolderResponseDescription>The Transaction was
approved</CardHolderResponseDescription>
   <MerchantResponseText>APPROVED</MerchantResponseText>
   <MerchantHelpText>The Transaction was approved</MerchantHelpText>
   <MerchantResponseDescription>The Transaction was
approved</MerchantResponseDescription>
   <UrlFail></UrlFail>
   <UrlSuccess></UrlSuccess>
   <EnablePostResponse>0</EnablePostResponse>
   <PxPayName></PxPayName>
   <PxPayLogoSrc></PxPayLogoSrc>
   <PxPayUserId></PxPayUserId>
```

```
    <PxPayXsl></PxPayXsl>
    <PxPayBgColor></PxPayBgColor>
    <PxPayOptions></PxPayOptions>
    <Cvc2ResultCode>P</Cvc2ResultCode>
    <AcquirerPort>100000000000-18270000</AcquirerPort>
    <AcquirerTxnRef>486310</AcquirerTxnRef>
    <GroupAccount>9997</GroupAccount>
    <DpsTxnRef>000000030884cdc6</DpsTxnRef>
    <AllowRetry>1</AllowRetry>
    <DpsBillingId></DpsBillingId>
    <BillingId></BillingId>
    <TransactionId>0884cdc6</TransactionId>
    <PxHostId>00000003</PxHostId>
    <RmReason></RmReason>
    <RmReasonId>0000000000000000</RmReasonId>
    <RiskScore>-1</RiskScore>
    <RiskScoreText></RiskScoreText>
</Transaction>
<ReCo>00</ReCo>
<ResponseText>APPROVED</ResponseText>
<HelpText>Transaction Approved</HelpText>
<Success>1</Success>
<DpsTxnRef>000000030884cdc6</DpsTxnRef>
 <TxnRef>inv1278</TxnRef>
</Txn>
```

Key Parameter	Description
AuthCode	Authorisation code given back from the bank for that transaction
Authorized	1 if transaction successful - 0 if declined or unsuccessful
Cvc2ResultCode	CVC / CVV2 Result Code associated with the result of the CVC validation
DateSettlement	Date transaction will be settled to Merchant Bank Account in YYYYMMDD format
DpsTxnRef	Required for refund and complete transactions.
DpsBillingId	Contains the BillingId generated by Payment Express when adding a card for recurring billing.
HelpText	A more detailed explanation of the response from the bank
ReCo	2 character response code
ResponseText	Response Text associated with ResponseCode
StatusRequired	1 if the result of the transaction could not be determined. See the Exception Handling section.
Success	1 if transaction successful - 0 if declined or unsuccessful. Will be the same value as Authorized

EXCEPTION HANDLING

This allows checking the Status of a Txn.

If you didn't receive a response to your POST or if StatusRequired was set to 1 in the response then you must send another Post to request the status of the transaction. For this function to work, you will need to send a TxnId with your original transaction. TxnId must be a unique value for each transaction. It can be up to 16 characters long.

XML Format of a transaction status Post:

```
<Txn>
<PostUsername>Sample</PostUsername>
<PostPassword>test1234</PostPassword>
<TxnType>Status</TxnType>
<TxnId>inv1278</TxnId>
</Txn>
```

or

```
<Txn>
<PostUsername>Sample</PostUsername>
<PostPassword>test1234</PostPassword>
<TxnId>inv1278</TxnId>
</Txn>
```

WELL FORMED XML

Character data sent via PX Post must be well formatted XML. For example, the following is invalid XML:

```
<Txn>
<CardHolderName>Bill & Son</CardHolderName>
<MerchantReference>Abc >> 123</MerchantReference>
</Txn>
```

Payment Express will be unable to read this XML and will return an error. If there is a possibility that a value will contain invalid characters (such as '&' in the cardholder name), please format the value using "HtmlEncoding".

The above example should be formatted as follows:

```
<Txn>
<CardHolderName>Bill &amp; Son</CardHolderName>
<MerchantReference>Abc &gt;&gt; 123</MerchantReference>
</Txn>
```

ELEMENT DESCRIPTIONS

Amount (input) Datatype: String Max 13 characters
Total Purchase, Refund, Auth or Completion amount. Format is d.cc where d is dollar amount (no currency indicator) and cc is cents amount. For example, $1.80 (one dollar and eighty cents) is represented as "1.80", not "1.8". A string value is used rather than the conventional Currency Datatype to allow for easy integration with Web applications. The current maximum value allowable is $99,999.99. Note that acquirer or card limits may be lower than this amount. When submitting transactions for currencies with no decimal division of units such as JPY, the AmountInput must be in an appropriate format e.g. "10".

AuthCode (input) Datatype: String Max 22 characters
Authorization code returned for approved transactions.

Authorized (output) Datatype: Boolean
Indicates if the transaction was authorized or not. Either False (0) or True (1)

BillingId (input) Datatype: String Max 32 characters
If a token based billing transaction is to be created, a BillingId has to be supplied. This is an identifier generated by the merchant application that is used to identify a customer or billing entry and can be used as input instead of card number and date expiry for subsequent billing transactions. To add a BillingId in the transaction request the EnableAddBillCard element needs to be present and set to 1 (true). Upon rebilling this will need to be set to 0 (false).

CardHolderName (input)Datatype: String Max 64 characters
The cardholder name as it appears on customer card. Optional and may be left blank.

CardNumber (input) Datatype: String Max 20 characters
The card number. No leading or embedded blanks are permitted. Must contain a numeric value.

Cvc2 (input) Datatype: String Max 4 characters
Card Verification Code 2 number. Some payment cards are issued with additional identifying information. These cards will have the account number printed on the signature panel of the card followed by a three or four digit value. This value is generated by the issuing bank and can be verified by the bank. Payment card brands have varying names for the value:

American Express: Four-digit batch code (4DBC)
MasterCard: Card Verification Code 2 (CVC2)
Visa: Card Verification Value 2 (CVV2)

Supplying this value provides an indication of that the person participating in a transaction had physical possession of the card at some point in time.

Cvc2Presence (input) Datatype: INT
CVC Presence Verification. Values are 0, 1, 2 & 9.

Merchant to send Payment Express a presence indicator within "Cvc2Presence" field in the transaction request to one of the below:
0 - You (MERCHANT) have chosen not to submit CVC
1 - You (MERCHANT) have included CVC in the Auth / Purchase
2 - Card holder has stated CVC is illegible.
9 - Card holder has stated CVC is not on the card.

CVC2ResultCode (output) Datatype: String
The CVC result code indicate the following:

RESPONSE CODE	DEFINITION	DEFINITION
M	CVC matched.	You will want to proceed with transactions for which you have received an authorisation approval. A CVC match indicates the values provided matches the Issuing Banks details
N	CVC did not match.	You may want to follow up with the cardholder to verify the CVC value before completing the transaction, even if you have received an authorisation approval. The CVC details provided by the Cardholder do not match their Issuing Banks details
P	CVC request not processed.	Issuing Bank is unable to process CVC at this time
S	CVC should be on the card, but merchant has sent code indicating there was no CVC.	You may want to follow up with the cardholder to verify that the customer checked the correct location for the CVC. If the transaction is Approved you may also wish to consider not fulfilling the transaction
U	Issuer does not support CVC.	The card Issuing bank does not support CVC validation.
NotUsed	Issuer does not support CVC.	The card Issuing bank does not support CVC validation.

DateExpiry (input) Datatype: String Max 4 characters
Indicates card expiry date. Format is MMYY where MM is month 01-12 and Year 00-99. do not insert "/" or other delimiter.

DateSettlement (output) Datatype: String Max 8 characters
Indicates Date of settlement (when money will be deposited in Merchant bank account) if this is supported by the Acquirer, otherwise contains the date the transaction was processed in YYYYMMDD format.

DpsBillingId(input)Datatype: String Max 16 characters
When output, contains the Payment Express generated BillingId. Only returned for transactions that are requested by the application with the EnableAddBillCard value set to 1 (true) indicating a token billing entry should be created.

DpsTxnRef (input/output) Datatype: String Max 16 characters
Returned for every transaction. If the transaction was approved, DpsTxnRef can be used as input to a Refund transaction. Used to specify a transaction for refund without supplying the original card number and expiry date. The DpsTxnRef value returned by the original approved Auth transaction must be supplied also when doing a complete transaction.

EnableAddBillCard (input) Datatype: Boolean
To automatically add a card for subsequent billing purposes, set this to 1 (true). When generating a Billing Transaction for a previously loaded BillingId or DpsBillingId, EnableAddBillCard must be 0 (false).

DateStart (input) Datatype: String Max 4 characters
The Issue date of the customer's credit card, if Issuer requires this field to be present.
Format is MMYY where MM is month 01-12 and Year 00-99. do not insert "/" or other delimiter.
Used for Maestro/Solo cards.

InstallmentNumber (input) Datatype: Number
Number value that is used to indicate the current payment number for an installment transaction. For example, if the consumer is making payment 1 of 12, then this value should be set to 1. Only used for installment based payments.

InstallmentCount (input) Datatype: Number
Number value is used to indicate the total number of payments for an installment transaction. For example, if the consumer is making 12 installment payments for total payment, then this value should be set to 12. Only used for installment based payments.

DebtRepaymentIndicator (input) Datatype: Number
Only send this field and set it to 1 to indicate that a debt repayment transaction is to be processed. Otherwise it will be set to 0 by default.

IssueNumber (input) Datatype: INT
The Issue Number of your credit card if Issuer requires this field to be present.

EnableAvsData (input) Datatype: INT
Address Verification System property. Values are 1 (Enable Verification), 0 (Disable Verification). Your bank may require that you use AVS, in which case you will need to set to 1.

AvsPostCode (input) Datatype: String Max 20 characters
Address Verification System property. Post Code that is listed on the customer's bank statement.

AvsStreetAddress (input) Datatype: String Max 60 characters
Address Verification System property. Address that is listed on the customer's bank statement.

AvsAction (input) Datatype: INT
Address Verification System property. Values are 0, 1 & 2.

0 - Don't check AVS details with acquirer, but pass them through to Payment Express only.
1 - Attempt AVS check. If the acquirer doesn't support AVS or is unavailable, then transaction will proceed as normal. If AVS is supported it will check the transaction and give the result.
2 - The transactions needs to be checked by AVS, even if isn't available, otherwise the transaction will be blocked.
3 - AVS check will be attempted and any outcome will be recorded, but ignored i.e. transaction will not be declined if AVS fails or unavailable.

The value will most likely be 1 for most circumstances.

InputCurrency (input) Datatype: String Max 4 characters
Indicates currency used for this transaction. If blank, currency will be determined by the bank account used which is selected using the Username/Password details. Not all acquirers can support multiple currencies. All other banks can only transact in their home currency. Valid values for Currency are:

CAD	Canadian Dollar
CHF	Swiss Franc
DKK	Danish Krone
EUR	Euro
FRF	French Franc
GBP	United Kingdom Pound
HKD	Hong Kong Dollar
JPY	Japanese Yen
NZD	New Zealand Dollar
SGD	Singapore Dollar

THB	Thai Baht
USD	United States Dollar
ZAR	Rand
AUD	Australian Dollar
WST	Samoan Tala
VUV	Vanuatu Vatu
TOP	Tongan Pa'anga
SBD	Solomon Islands Dollar
PGK	Papua New Guinea Kina
MYR	Malaysian Ringgit
KWD	Kuwaiti Dinar
FJD	Fiji Dollar

MerchantReference (input) Datatype: String Max 64 characters
Free text to appear on transaction reports.

PostPassword (input) Datatype: String Max 32 characters
Used with PostUsername to determine account for settlement. Payment Express clients can be set up with more than one bank account. Each transaction may be designated for a specific account if required.

PostUsername (input) Data type: String Max 32 characters
Used with PostPassword to determine account for settlement. Payment Express clients can be set up with more than one bank account. Each transaction may be designated for a specific account if required.

ReceiptEmail (input) Datatype: String Max 64 characters
Store a card holder or customer's email address on the transaction details. It will be returned in the transaction response. The API account can be configured by PX Support to send out a transaction confirmation email automatically.

ReCo (Response Code) (output) Datatype: String Max 2 characters
The client application should not interpret the Response Code property contents - it is provided as informational only. The Authorized field in the response determines if the transaction was successful or not.

RecurringMode (input) Datatype: String
In the RecurringMode field please set the specific string value required to indicate the card storage reason when tokenising and rebilling a card with a token.
The string value depends on the business case when tokenising and rebilling a card. Please see Token Billing section for possible string values and their usage details.

RmReason (output) Datatype: String Max 255 characters
Risk management rule response Text

RmReasonId (output) Datatype: String Max 16
Risk Management Rule ID

RiskScore (output) Datatype: Int
The risk score associated with the transactions.

RiskScoreText (output) Datatype: String Max 2048 characters
The risk score text provides some meaningful text about the rick management rules that were hit and whether they caused the transaction to be blocked. Each rule is separated by a ','.These are followed by the credit card country of the cardholder and the IP Country of the cardholder.
RM Rule1 Hit Not Blocked, RM Rule2 Hit Blocked, CCC=NZ IPC=AU

StatusRequired (output) Datatype: Boolean
1 - If transaction result is unknown
0 - If transaction result is in the response. See the Exception Handling section.

Success (output) Datatype: Boolean
1 - If transaction successful
0 - If declined or unsuccessful. Will be the same value as Authorized

TxnData1, TxnData2, TxnData3 (input) Datatype: String Max 255 characters
Optional free text fields. Usually assigned at the merchant's/origin website.

TxnId (input/output) Datatype: String Max 16 characters
Input: contains a unique, merchant application generated value that uniquely identifies the transaction. If TxnId is used, you can check the status of a transaction. Where possible it is recommended that the merchant application sets this value.

TxnType (input) Datatype: String

Value	Meaning
Auth	Authorizes a transaction. Must be completed within 7 days using the "Complete" TxnType.
Complete	Completes (settles) a pre-approved Auth Transaction. The DpsTxnRef value returned by the original approved Auth transaction must be supplied.
Purchase	Purchase - Funds are transferred immediately.
Refund	Refund - Funds transferred immediately. Must be enabled as a special option.
Validate	Validation transaction. On the amount 0.00 or 1.00, validates card details including expiry date. Often utilised with the EnableAddBillCard property set to 1 to automatically generate a card's token for rebilling. Note that the Validate transaction type may not be enabled by default on live accounts. Please make a request to Payment Express Support if you would like to utilise this transaction type.

Track 2 DataType: String Max 37 characters
Extracted from Track2 of credit card. Numeric with an equal sign.
Example:
4111111111111111=08101011179517320000

When submitting encrypted track2 data the encrypted data sits between the ";" and "?" sentinels. Example:
;SaV_[va+tZnt1111=FG5Gv-.wM:Ypt#s].n01?

AUTH-COMPLETION

OVERVIEW

Payment Express supports Auth/Completion. An "Auth" transaction verifies that funds are available for the requested card and amount and reserves the specified amount. A "Completion" transaction is sent at a later date to cause funds transfer for the previously authorised amount, or a smaller amount if the total original value is no longer required. This transaction set is useful when the merchant needs to ensure that funds up to a certain limit are available but the actual total amount is not yet known or goods or services have not yet been delivered.

OPERATION

1) Authorization

Set TxnType to "Auth" for the amount to be authorised. The Auth response contains a DpsTxnRef. The funds are not transferred from the cardholder account.

2) Completion

After a successful Authorization transaction, but within 7 days maximum, a "completion" (TxnType="Complete") transaction must be sent containing the DpsTxnRef returned by the "Auth" transaction.

TOKEN BILLING

OVERVIEW

Token Billing allows for regular billing of a cardholder card, under the control of the merchant, without requiring the merchant to either store sensitive card data securely or to obtain credit card details every time a new payment is requested.

This functionality is implemented by proving the ability for a merchant to request payment express to capture and store credit card number and expiry date and to link these stored details to a merchant supplied "BillingId". The BillingId is a 32 character field that contains a reference that is unique to the merchant's customer that will be associated with the credit card information stored securely at Payment Express. This is undertaken during the Setup Phase. For subsequent charges to the card (Rebill Phase), the merchant does not need to supply the card number or expiry date, only the BillingId originally associated during the Setup Phase.

OPERATION

1) Setup Phase

The setup phase involves tokenising a card with the PxPost API. As the card is captured within the merchant hosted system, the integrated system's checkout process has to be PCI DSS SAQ Type D compliant or the integrated system may already be PCI Level 1 Compliant. This compliance may be audited by the acquirer or financial entity on go live.

The tokenising setup phase requests an online $0.00 Validate (TxnType: Validate) or $1.00 Authorisation (TxnType: Auth) transaction which will determine that the card is valid and not on hot or stolen card lists and that it has the valid expiry date. The TxnType: Purchase is used if the card is to be charged with an amount and tokenised at the same time.

Merchants will typically integrate directly into their ecommerce web site or application or call centre for the setup phase.

To store or add a card for future rebilling, send a new transaction request with the following properties:

- TxnType (Validate or Auth or Purchase)
- InputCurrency
- Amount
- TxnId (unique merchant system specified ID to uniquely identify the transaction)
- MerchantReference
- EnableAddBillCard (Set to 1 to store card)
- RecurringMode (required to indicate the reason for tokenising the card – see the below for use case description)
- CardNumber (required)
- DateExpiry (optional - strongly recommended)
- CardHolderName (optional - strongly recommended)
- Cvc2 (optional)
- Cvc2Presence (optional)
- BillingId (optional - included when saving the card as a token using the merchant system's specified Billing ID. Or, parse the DpsBillingId or CardNumber2 generated by Payment Express only in the Txn response)

In the **RecurringMode** request field, please set one of the card storage reason as the string listed below. When tokenising the card, please set one of the following:

RecurringMode	**Usage explanation**
credentialonfileinitial	Cardholder will save card and for future orders the cardholder selects to reuse the saved card for the one-off payment.
unscheduledcredentialonfileinitial	Cardholder will save their card and for future order based on an event (such as topup) the merchant will reuse the saved card on behalf of the cardholder for the one-off payment.
recurringinitial	Cardholder will save their card and merchant will reuse the saved card on behalf of cardholder for the subscribed recurring payments.
installmentinitial	Cardholder will save their card and merchant will reuse the saved card on behalf of cardholder for the installment payments.

Please discuss with our Implementation and Sales team about your tokenisation use cases if you are unsure. The RecurringMode string value should be set based on the merchant's business case for tokenising.

Sample Transaction request input to store the card as a token generated by Payment Express:

```
<Txn>
  <PostUsername>TestUsername</PostUsername>
  <PostPassword>TestPassword</PostPassword>
  <TxnType>Validate</TxnType>
  <InputCurrency>NZD</InputCurrency>
  <Amount>1.00</Amount>
  <TxnId>UniqueOrderID</TxnId>
  <MerchantReference>Test Transaction</MerchantReference>
  <CardNumber>4111111111111111</CardNumber>
  <CardHolderName>Sample Cardholder Name</CardHolderName>
  <DateExpiry>1010</DateExpiry>
  <Cvc2>345</Cvc2>
  <Cvc2Presence>1</Cvc2Presence>
  <EnableAddBillCard>1</EnableAddBillCard>
  <RecurringMode>credentialonfileinitial</RecurringMode>
</Txn>
```

Sample Transaction request input to tokenise the card with BillingId token generated by merchant's integrated solution:

```
<Txn>
  <PostUsername>TestUsername</PostUsername>
  <PostPassword>TestPassword</PostPassword>
  <TxnType>Validate</TxnType>
  <InputCurrency>NZD</InputCurrency>
  <Amount>1.00</Amount>
  <TxnId>UniqueOrderID</TxnId>
  <MerchantReference>Test Transaction</MerchantReference>
  <CardNumber>4111111111111111</CardNumber>
  <CardHolderName>Sample Cardholder Name</CardHolderName>
  <DateExpiry>1010</DateExpiry>
  <Cvc2>345</Cvc2>
  <Cvc2Presence>1</Cvc2Presence>
  <EnableAddBillCard>1</EnableAddBillCard>
  <BillingId>UniqueMerchantTokenPerCard</BillingId >
  <RecurringMode>credentialonfileinitial</RecurringMode>
</Txn>
```

2) Rebill Phase

To rebill a card with a token the merchant application requests a new transaction via PxPost request with following properties:

- TxnType (usually Purchase to rebill or charge the card with token)
- InputCurrency
- Amount (amount to rebill or charge)
- TxnId (unique merchant system specified ID to uniquely identify the transaction)
- MerchantReference
- RecurringMode (required to indicate the reason for rebilling the card token – see the below for use case description)
- DpsBillingId or BillingId or CardNumber2 (depending on system's preferred token rebill usage)

It is important to set the **RecurringMode** request field, please set one of the card storage reason as the string listed below.
When **rebilling the card with token**, please set one of the following:

RecurringMode	**Usage explanation**
credentialonfile	Cardholder selects their saved card to make the one-off rebill payment.
unscheduledcredentialonfile	Merchant initiated and event driven one-off rebilling with stored card (e.g. auto topups).
installment	Merchant initiated rebilling payments in installments with a stored card token.
incremental	Merchant initiated incremented transaction amount to rebill, e.g. hospitality, rentals, etc.
recurring	Merchant initiated recurring transaction with a stored card token (e.g. subscriptions).
recurringnoexpiry	Merchant initiated recurring transaction with a stored card token where no card expiry check needs to occur (e.g. subscriptions).
resubmission	Merchant resubmits rebill with token where it requested an authorisation, but may have received a decline due to insufficient funds, but the order already delivered to the cardholder. Used with the token to get an outstanding payment from cardholder.
reauthorisation	Merchant initiated when the completion or conclusion of the original order or service extends beyond the authorisation validity. Common for retail (split or delayed shipments) and hospitality or rental services scenarios.
delayedcharges	Merchant initiated to process additional account rebill charge after original order and payment has been already processed and fulfilled.
noshow	Merchant initiated to charge the cardholder a penalty relevant to the merchant's cancellation policy. Common for guaranteed reservations scenarios, e.g. Hospitality.

Please discuss with our Implementation and Sales team about your rebilling use cases if you are unsure. The RecurringMode string value should be set based on the merchant's business case for rebilling the card.

Once the rebill request is received Payment Express processes the token with the associated credit card number and expiry date stored in the Setup Phase and a purchase transaction is formatted and processed to the card acquirer. Once the acquirer has processed the transaction the PxPost transaction response is sent back.

Sample PxPost Transaction request input to rebill the card with DpsBillingId token generated by Payment Express from the tokenising setup phase:

```
<Txn>
  <PostUsername>TestUsername</PostUsername>
  <PostPassword>TestPassword</PostPassword>
  <TxnType>Purchase</TxnType>
  <InputCurrency>NZD</InputCurrency>
  <Amount>100.00</Amount>
  <TxnId>UniqueOrderID</TxnId>
  <MerchantReference>Test Transaction</MerchantReference>
  <DpsBillingId>0000060004444444</DpsBillingId>
  <RecurringMode>credentialonfile</RecurringMode>
</Txn>
```

Alternatively the rebilling requests can be processed via Payline Batch Processor or SFTP Batch Processor.

CardNumber2 Usage

CardNumber2 is a token generated by Payment Express and associated with card details supplied. It is 16 numeric characters and conforms to a Luhn "mod 10" algorithm. This makes it ideal for storage within the database in place of a card number where the value is validated against checks which might normally be made against credit card numbers. A CardNumber2 value is always unique for a given card number. Should a card number be presented for tokenization multiple times the same CardNumber2 value will be returned.

CardNumber2 tokens are generated for all transactions once enabled by Payment Express (please contact your Payment Express account manager to discuss). The token number will be returned in the "CardNumber2" property of the transaction result.

Charging a CardNumber2 token involves a request from the merchant application or Batch processor including an appropriate cardNumber2, a TxnType (Purchase) and the amount to be charged (an optional MerchantReference can be added for reporting purposes). EnableAddBillCard value will need to be set to "False" (or 0) for the rebill phase. Payment Express® retrieves the credit card number and expiry date stored in the Setup Phase and a purchase transaction is formatted and processed to the card acquirer.

CardNumber2 transactions use the card expiry date stored with the token regardless of whether one is passed through in the transaction data. Once a successful transaction is processed using the real card number associated with a CardNumber2 token the expiry date stored with this token will be updated to that which was used to process the transaction. If your client application displays details of stored tokens to cardholders e.g.: masked number and expiry date, then it is advisable upon a successful transaction for the merchant application to update the expiry date that is stored with the generated token.

REFUNDS

PxPost is capable of handling refunds (credit) transactions; however you will need to match the original Purchase or Complete transaction for this to happen. The matching is done with the DpsTxnRef given from the response of a purchase or complete transaction. You are able to do multiple refund transactions to the maximum amount of the original matched transaction.

The TxnType will be Refund.

The Payment Manager is provided to merchants with all integrated solutions by Payment Express, so there is a ready built interface to handle refund transactions already. However, if you wish to integrate refunds into your own interfaces the following input properties need to be provided for a refund transaction:

TxnType = Refund
DpsTxnRef
MerchantReference
Amount

```
<Txn>
<PostUsername>TestUsername</PostUsername>
<PostPassword>TestPassword</PostPassword>
<Amount>1.23</Amount>
<TxnType>Refund</TxnType>
<DpsTxnRef>0000000400000000</DpsTxnRef>
<MerchantReference>Refund Order</MerchantReference>
</Txn>
```

RESPONSE CODES

The client application should not interpret the ReCo (Response Code) property contents - it is provided as information only. The Success property determines if the the transaction was successful or not.

Troubleshooting Errors

The following table provides assistance in troubleshooting errors:

Error Code	Explanation
D2	No such user for PXPost. Please contact Payment Express to confirm your account information.
D3	Blank password for PX Post. Please contact Payment Express to confirm your account information.
D5	Invalid Password for PxPost. Please contact Payment Express to confirm your account information.