



Payment Express Hosted – PX Pay 2.0 Integration Guide

Version 2.7

COPYRIGHT

© Copyright 2019, Payment Express
98 Anzac Avenue
PO Box 8400
Auckland, 1150
New Zealand
www.paymentexpress.com

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the express written permission of Payment Express.

PROPRIETARY NOTICE

The information described in this document is proprietary and confidential to Payment Express. Any unauthorised use of this material is expressly prohibited except as authorised by Payment Express in writing.

CONTENTS

| | |
|---|----|
| Copyright..... | 2 |
| Proprietary Notice | 2 |
| OVERVIEW | 5 |
| Key features | 5 |
| How It Works..... | 6 |
| BASIC COMMUNICATION | 7 |
| INTEGRATION METHODS..... | 8 |
| Mobile Device | 8 |
| Redirect | 8 |
| iFrame – Embedded Hosted Payment Page | 9 |
| PREPARATION | 10 |
| PX Pay 2.0 development account..... | 10 |
| PX Pay 2.0 sample code | 10 |
| TRANSACTION REQUEST | 11 |
| GenerateRequest XML Document | 11 |
| Request XML Document..... | 12 |
| TRANSACTION RESPONSE..... | 13 |
| ProcessResponse XML Document | 13 |
| Response XML Document | 14 |
| ELEMENT DESCRIPTIONS | 15 |
| COMMON SCENARIOS | 20 |
| Purchase Transaction Example..... | 20 |
| Auth Transaction Example..... | 22 |
| Finalizing Auth Transactions..... | 24 |
| Option 1: PX Pay 2.0 > Payline..... | 25 |
| Option 2: PX Pay 2.0 > Batch Upload | 26 |
| Option 3: PX Pay 2.0 > Batch Processor | 26 |
| Option 4-5: PX Pay 2.0 > PX Post/Webservice..... | 27 |
| TOKEN BILLING | 28 |
| Setup Phase | 28 |
| Token Creation Example..... | 29 |
| Rebill Phase | 32 |
| Option 1: PX Pay 2.0..... | 33 |
| Option 2: PX Pay 2.0 > Batch Upload | 36 |
| Option 3: PX Pay 2.0 > Batch Processor | 37 |
| Option 4-5: PX Pay 2.0 > PX Post or Webservice..... | 37 |
| UPOP (UnionPay Online Payments)..... | 38 |
| HPP CUSTOMISATION | 44 |
| HPP Customisation via Payline | 45 |
| HPP Page 1 Sample - Customisable Fields (Payment Details & Payment Page)..... | 46 |
| HPP Page 1 Sample - Customisable Fields (Page Style) | 46 |
| HPP Page 2 Sample - Customisable Fields (Result Page)..... | 48 |

| | |
|---|----|
| Merchant submitted CSS..... | 49 |
| FAIL-PROOF RESULT NOTIFICATION (FPRN) | 50 |
| 3D SECURE..... | 51 |
| TROUBLESHOOTING & FAQs | 52 |
| REQUEST RESPONSE CODES | 53 |
| GO LIVE..... | 54 |

OVERVIEW

The PX Pay 2.0 interface is a platform independent Hosted Payments Page (HPP) solution provided by Payment Express. The HPP provides a solution for the capturing credit card information securely without exposing the merchant to the sensitive data.

This is achieved by allowing the card holder to enter their card details into a page which is hosted by Payment Express rather than the merchants own website. The major advantage of this approach is that the merchant does not see, and is not aware of, the card number at any point in the process. This is beneficial from a PCI DSS standpoint because the scope of PCI DSS requirements is likely to be reduced.

PCI DSS (Payment Card Industry Data Security Standard) is a set of comprehensive requirements created by card issuers American Express, Discover Financial Services, JCB International, MasterCard and Visa to ensure the security of credit card data online. All merchants, whether small or large, need to be PCI compliant. Payment Express is registered as a PCI DSS compliant service provider; therefore a payment page solution hosted by Payment Express meets all PCI DSS requirements.

KEY FEATURES

A demonstration of PX Pay 2.0 can be found online at <https://demo.paymentexpress.com/SandboxMain.aspx>



Payment Checkout

Merchant Reference: Sample Merchant Reference
Order Line 1: Sample TxnData1
Amount: \$ 1.00 (NZD)

Select Payment Method

Select a payment method from the following available options:

Credit Card
 Account2Account

Credit Card Payment

Card Number:*
Card Holder Name:*
Expiry Date (MM/YY):*
Card Security Code:*

Cancel Payment Submit

paymentexpress
Privacy Policy

Payment Checkout

Merchant Reference: Sample Merchant Reference
Order Line 1: Sample TxnData1
Amount: \$ 1.00 (NZD)

Select Payment Method

Select a payment method from the following available options:

Credit Card
 Account2Account

Credit Card Payment

Card Number:*
Card Holder Name:*
Expiry Date (MM/YY):*
Card Security Code:*

Cancel Payment Submit

paymentexpress
VISA MasterCard American Express JCB Discover UnionPay WeChat Pay

Privacy Policy

Payment Checkout

Merchant Reference: Sample Merchant Reference
Order Line 1: Sample TxnData1
Amount: \$ 1.00 (NZD)

Select Payment Method

Select a payment method from the following available options:

Credit Card
 Account2Account

Credit Card Payment

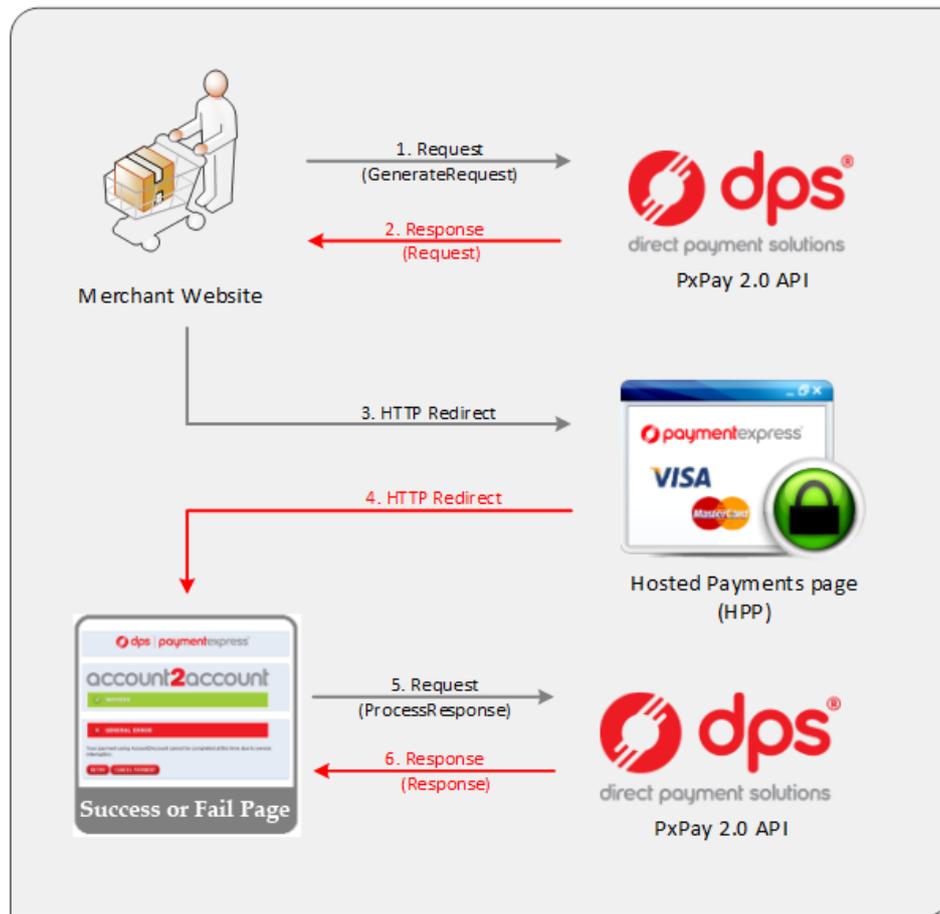
Card Number:*
Card Holder Name:*
Expiry Date (MM/YY):*
Card Security Code:*

Cancel Payment Submit

paymentexpress

HOW IT WORKS

PxPay 2.0 Process Flow



1. To process a transaction, PX Pay 2.0 allows merchants to send XML requests to Payment Express® via HTTPS posts to <https://sec.paymentexpress.com/pxaccess/pxpay.aspx>. PxPay Username & PxPay Key is required too. For testing PxPay 2.0 on our UAT environment - please send XML requests to <https://uat.paymentexpress.com/pxaccess/pxpay.aspx>.
2. Payment Express® responds with a unique URI (encrypted URL) for an SSL secure payments page.
3. The merchant shopping cart uses the returned URI to redirect the customer to the secure Payment Express hosted payments page.
4. The customer will be prompted to enter their credit card details and complete the transaction. The transaction is then sent to the merchant bank for authorisation. The result is displayed and the user is automatically redirected back to the merchant's website (success or fail page);
i.e.
<https://demo.paymentexpress.com/SandboxSuccess.aspx?result=0000840000185376f1519ff80a5ccd54&userid=SampleUserId>
5. You take the "result" parameter value in the URL string i.e. 0000840000185376f1519ff80a5ccd54 along with the PX Pay username and PX Pay key; to send the response request (ProcessResponse) to Payment Express and receive the XML response back.
6. The transaction results and other transaction details are decrypted and sent back to the merchant as a standard XML response. **NOTE:** In case a blank XML response is returned, please retry the ProcessResponse twice with an interval of 2 seconds.

BASIC COMMUNICATION

Character data sent via PX Pay 2.0 must be well formed XML.

The XML document must contain the required **opening** and **closing tags** that contains the whole document i.e. the root element.

Example: When generating the input XML document to begin a transaction request, the following GenerateRequest opening and closing tags must be present.

```
<GenerateRequest>
...
</GenerateRequest>
```

All tags must be **nested properly**. There must be an opening and a closing tag for all elements and the tags cannot overlap.

Example: Closing tags not complete.

`</AmountInput` - has no closing angle bracket, therefore the tag is not complete.

`</AmountInput)` - has a wrong closing bracket, therefore the tag is not complete.

The XML tags are **case sensitive** and **unique**. If a tag is submitted which is not recognized by Payment Express and is not a required element, it will be ignored and will not be returned in the response. If the tag is for a required element, an error may occur and a response code will be returned.

Example: If the AmountInput tag was sent with a lowercase "i" instead of an uppercase "I" and error will occur the response code "IU – Invalid Amount" will be returned.

`<Amountinput>1.00</Amountinput>` - Incorrect

`<AmountInput>1.00</AmountInput>` - Correct

If there is a possibility that a value will contain **invalid characters** (such as '&' and '>' in the cardholder name), please format the value using "HtmlEncoding", otherwise Payment Express will be unable to read the XML and will return an error (i.e. "Not acceptable input XML").

Example, the following is invalid XML:

```
<GenerateRequest>
<TxnData1>Bill & Son</TxnData1>
<MerchantReference>Abc >> 123</MerchantReference>
</GenerateRequest>
```

The following is how it should be formatted.

```
<GenerateRequest>
<TxnData1>Bill &amp; Son</TxnData1>
<MerchantReference>Abc &gt;&gt; 123</MerchantReference>
</GenerateRequest>
```

INTEGRATION METHODS

Generally merchants implement a Payment Express hosted payment page solution in one of two ways; either redirecting the user and their entire browser to the payment page or by presenting the payment page within an inline frame embedded in a page on their website.

MOBILE DEVICE

The PxPay 2.0 interface can also be integrated on a native mobile application as a payment method. The mobile application may utilise a webpage component to view the hosted payment page over HTTPs. The mobile platform being integrated with PxPay 2.0 should support HTTPS posts and XML data exchange.

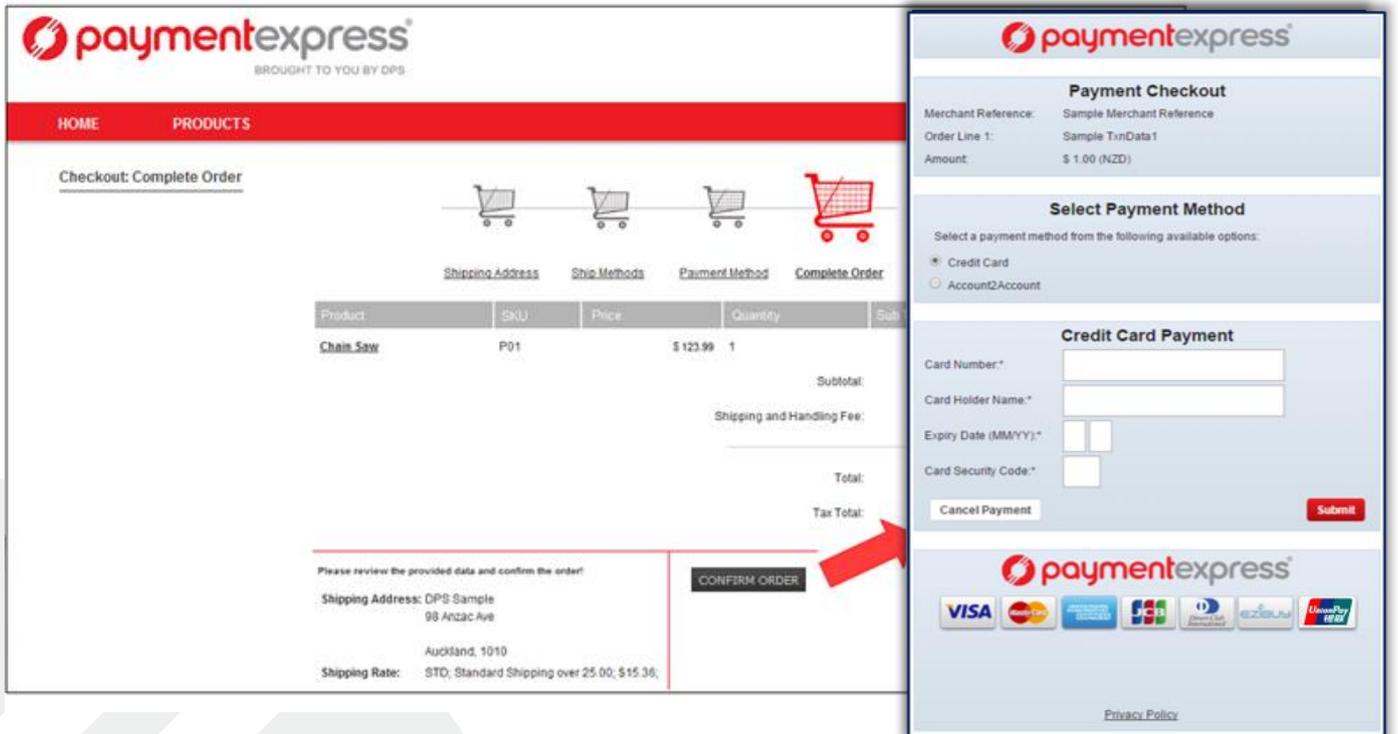
The PxPay 2.0 payment automatically switches to a mobile format for mobile devices by default. If the PxPay 2.0 payment page is not responsive to the mobile device's screen as expected, please ensure the user agent string that the device's web browser is reporting is mobile specific within the first 255 characters of the string.

If the hosted payment page is still not responsive on the mobile device screen, please note the following:

- Please email devsupport@paymentexpress.com and quote the exact user agent string of the relevant mobile device(s) used to send the transaction request.
- Also if a mobile web browser is used to redirect to the hosted payment page, please specify the exact mobile web browser and the version.

REDIRECT

The redirect integration method involves directing the user away from the merchant website to a Payment Express-hosted page for the purposes of collecting credit card details. Once credit card details have been collected and a transaction processed the user is directed back to the merchant website. The image below demonstrates a payment page accessed using the redirection method.

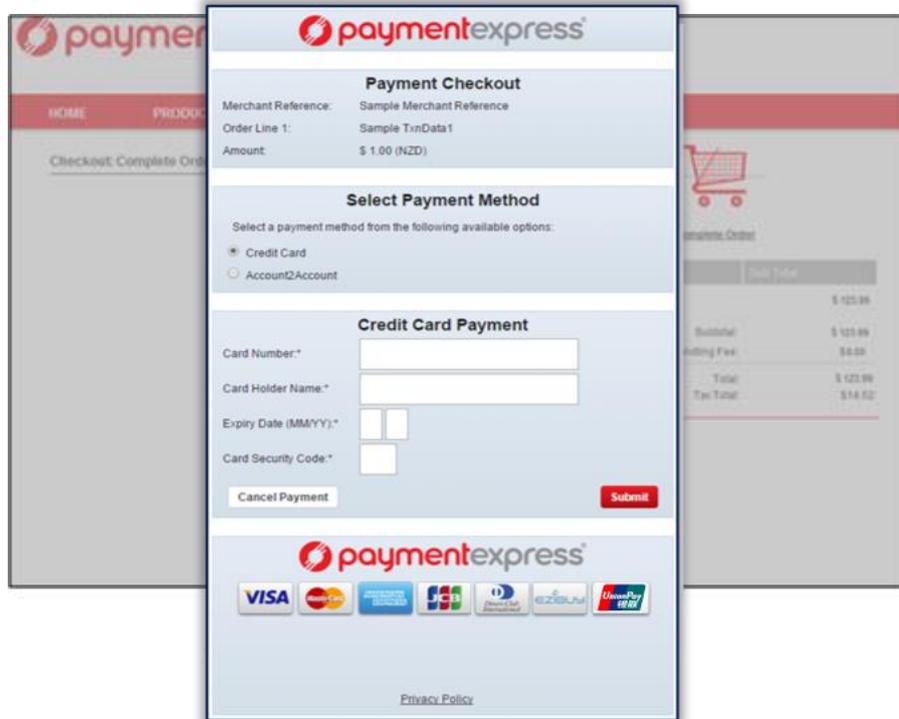


Example Merchant Website

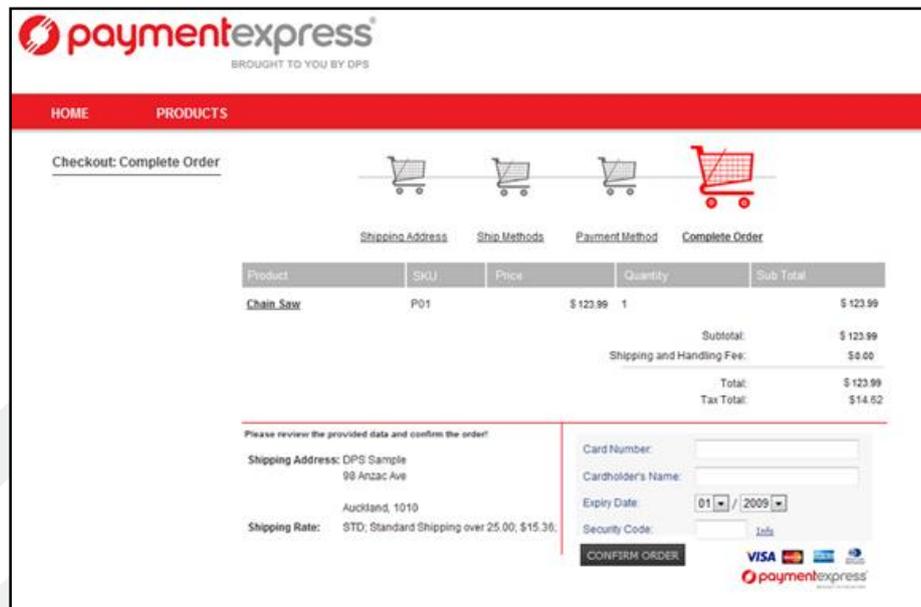
Payment Express Hosted Payments Page (HPP)

IFRAME – EMBEDDED HOSTED PAYMENT PAGE

The iframe integration method involves presenting the Payment Express hosted payment page within the merchant website inside a frame. The iframe content can either be presented as the page loads or asynchronously (outside the normal page request flow) based upon user interaction. Note that this method of integration may increase the scope of applicable PCI-DSS requirements. Please speak to your acquirer to confirm their position on this particular implementation of the Payment Express hosted payment page. The images below demonstrate the iframe method of integration.



Payment Express Hosted Payments Page (HPP) embedded in an iframe (ThickBox Example)



Payment Express Hosted Payment Page (HPP) embedded in an iframe

PREPARATION

To begin integration testing you will need the following:

- Payment Express® PX Pay 2.0 development accounts – Contact our Ecommerce sales team to request a Dev account (Please refer to: http://www.paymentexpress.com/Contact_Us), or apply online at <https://sec.paymentexpress.com/pxmi/apply>.
- PX Pay 2.0 interface technical specification - www.paymentexpress.com/Technical_Resources/Ecommerce_Hosted/PxPay_2_0
- PX Pay 2.0 sample code - www.paymentexpress.com/Technical_Resources/Ecommerce_Hosted/PxPay_2_0

PX PAY 2.0 DEVELOPMENT ACCOUNT

A PX Pay 2.0 development account is usually setup within 1-3 business days. Each test account will be assigned to the Payment Express test environment which simulates a connection to the merchant bank. To access the PX Pay 2.0 account, a UserId and Key will be provided.

Example: **PxPayUserId:** Sample2_Dev
PxPayKey: abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890

All PX Pay 2.0 accounts also come with a Payline account. Developers can use Payline to track down their test transactions, process transactions manually, and generate transaction reports. To access the Payline account, use the PxPayUserId along with a unique alphanumeric password setup just for Payline.

Payline login URL: <https://sec.paymentexpress.com/pxmi3/logon>.

Example: **Payline Username:** Sample2_Dev (Same as PxPay2UserId)
Payline Password: abcd1234

PX PAY 2.0 SAMPLE CODE

Sample code can be provided in the following languages:

- PHP cURL
- PHP OpenSSL
- ASP.Net 3.5 (C#)
- ASP.Net 3.5 (VB)
- Java
- ColdFusion

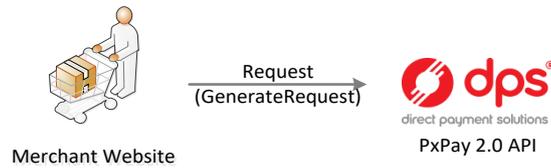
You will need to ensure the client library sending HTTP request is using TLSv1.2 for connectivity at all times.

All the sample codes can be downloaded from the link:

https://www.paymentexpress.com/Downloads/PxPay2_SampleCode.zip

TRANSACTION REQUEST

GenerateRequest XML Document – To initiate a transaction the merchant posts the GenerateRequest to <https://sec.paymentexpress.com/pxaccess/pxpay.aspx>



```
<GenerateRequest>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<TxnType>Purchase</TxnType>
<AmountInput>1.00</AmountInput>
<CurrencyInput>NZD</CurrencyInput>
<MerchantReference>Purchase Example</MerchantReference>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<EmailAddress>SampleUserId@paymentexpress.com</EmailAddress>
<TxnId>ABC123</TxnId>
<BillingId>BillingId123xyz</BillingId>
<EnableAddBillCard>1</EnableAddBillCard>
<RecurringMode>single</RecurringMode>
<UrlSuccess>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlSuccess>
<UrlFail>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlFail>
<UrlCallback>https://InsertValidUrlForCallback</UrlCallback>
</GenerateRequest>
```

Note: Elements in blue text are optional

The following is a list of the inputs elements applicable for a GenerateRequest.

GenerateRequest (Input XML Document)

| Input Element | Required | Datatype |
|-------------------|----------|-------------------|
| PxPayUserId | Yes | Max 32 bytes |
| PxPayKey | Yes | Max 64 bytes |
| AmountInput | Yes | Max 13 characters |
| BillingId | No | Max 32 characters |
| CurrencyInput | Yes | Max 4 characters |
| EmailAddress | No | Max 255 bytes |
| EnableAddBillCard | No | Max 1 character |
| RecurringMode | No | String |
| MerchantReference | No | Max 64 bytes |
| DpsBillingId | No | Max 16 characters |
| TxnData1 | No | Max 255 bytes |
| TxnData2 | No | Max 255 bytes |
| TxnData3 | No | Max 255 bytes |
| TxnType | Yes | Max 8 Characters |
| TxnId | Yes | Max 16 bytes |
| UrlFail | Yes | Max 255 bytes |

| | | |
|------------------------|-----|---------------|
| UrlSuccess | Yes | Max 255 bytes |
| UrlCallback | No | Max 255 bytes |
| Opt | No | Max 64 bytes |
| ClientType | No | String |
| ForcePaymentMethod | No | String |
| DebtRepaymentIndicator | No | Number |
| InstallmentNumber | No | Number |
| InstallmentCount | No | Number |

Request XML Document - Once the GenerateRequest has been processed a Request will be returned.



The URI returned can then be used to redirect the customer to the Payment Express Hosted Payments Page.

The following is a list of the output elements applicable for a Request.

```
<Request valid="1">
<URI>https://sec.paymentexpress.com/pxmi3/EF4054F622D6C4C1B4F9AEA59DC91CAD3654CD60ED7ED04110CBC402959AC7
CF035878AEB85D87223</URI>
</Request>
```

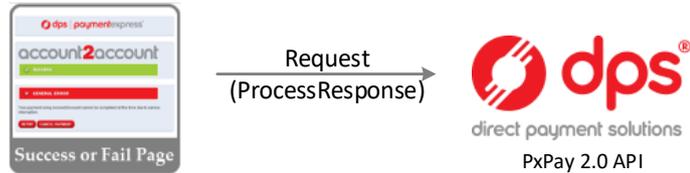
Request (Output XML Document)

| Output Element | Datatype |
|-------------------|-------------|
| valid [Attribute] | 1 character |
| URI | String |

TRANSACTION RESPONSE

ProcessResponse XML Document – Once the user has submitted their credit card information and the transaction has been processed, the merchant's UriSuccess or UriFail or UriCallback (if set in the request) will receive an [FPRN](#) with the GET HTTP message with the result value that needs to be extracted and requested as below to obtain the transaction outcome and details.

To extract the transaction details from the encrypted URI string, the merchant sends the encrypted result string as Response value with their PXPAYUsername and PXPAYKey in the ProcessResponse (Input XML Document).



The following is a list of the input elements applicable to the ProcessResponse. A full description of all available elements can be found on [Element Descriptions](#) section.

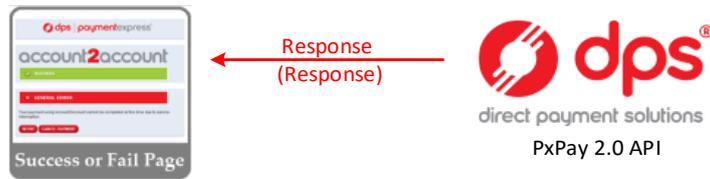
. Example:

```
<ProcessResponse>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<Response>0000840000185376f1519ff80a5ccd54</Response>
</ProcessResponse>
```

ProcessResponse (Input XML Document)

| Input Element | Required | Datatype |
|---------------|----------|--------------|
| PxPayUserId | Yes | Max 32 bytes |
| PxPayKey | Yes | Max 64 bytes |
| Response | Yes | String |

Response XML Document - The Response will return the decrypted transaction results for the merchant to interpret.



The following is a list of the output elements applicable to the Response. A full description of all available elements can be found on [Element Descriptions](#) section.

Response (Output XML Document)

| Output Element | Datatype | Output Element | Datatype |
|--------------------|-------------------|-------------------|-------------------|
| valid [Attribute] | 1 character | TxnData1 | Max 255 bytes |
| AmountSettlement | Max 13 characters | TxnData2 | Max 255 bytes |
| AuthCode | Max 22 characters | TxnData3 | Max 255 bytes |
| CardName | Max 16 bytes | TxnType | Max 8 Characters |
| CardNumber | Max 16 bytes | CurrencyInput | Max 4 characters |
| DateExpiry | Max 4 characters | MerchantReference | Max 64 bytes |
| DpsTxnRef | Max 16 bytes | ClientInfo | Max 15 characters |
| Success | Long | TxnId | Max 16 bytes |
| ResponseText | Max 32 bytes | EmailAddress | Max 255 bytes |
| DpsBillingId | Max 16 characters | BillingId | Max 32 characters |
| CardHolderName | Max 64 bytes | TxnMac | Max 32 characters |
| CurrencySettlement | Max 4 characters | CardNumber2 | Max 16 characters |
| PaymentMethod | String | Cvc2ResultCode | 1 character |

ELEMENT DESCRIPTIONS

PxPayUserId (input) Datatype: BSTR Max 32 bytes

The PxPayUserId is a unique username to identify your customer account. This username will be setup by our Payment Express® Activations team and issued with your development account. The standard format used for Dev PxPayUserId is 'merchantname_dev' e.g. storename_dev.

PxPayKey (input) Datatype: BSTR Max 64 bytes

The PxPayKey is a unique 64 character key to identify customer account and used to encrypt the transaction request with 3DES to protect the transaction information. This key will be setup by our Payment Express® Activations team and issued with your development account.

AmountInput (input) Datatype: BSTR Max 13 characters

The AmountInput field specifies the total Purchase or Auth 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 maximum value allowable is "999999.99" however acquirer or card limits may be lower than this amount.

When submitting transactions for currencies with no decimal division of units such as the Japanese Yen (JPY), the AmountInput must be in an appropriate format e.g. "1000" for 1000 Yen. If "1000.00" is submitted in AmountInput with CurrencyInput set to JPY, an error will occur and response code "IU" will be returned.

BillingId (input/output) Datatype: BSTR Max 32 characters

If a token based billing transaction is to be created, a BillingId may be supplied. This is an identifier supplied 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. Token Billing is discussed in detail on [Token Billing Section](#).

CurrencyInput (input/output) Datatype: BSTR Max 4 characters

The CurrencyInput field is used to specify the currency to be used e.g. "NZD" or "AUD". Possible range of currencies:

| | | | | | |
|-----|----------------------|-----|-------------------------|-----|----------------------|
| AUD | Australian Dollar | HKD | Hong Kong Dollar | SGD | Singapore Dollar |
| BND | Brunei Dollar | INR | Indian Rupee | THB | Thai Baht |
| CAD | Canadian Dollar | JPY | Japanese Yen | TOP | Tongan Pa'anga |
| CHF | Switzerland Franc | KWD | Kuwait Dinar | USD | United States Dollar |
| EUR | Euro | MYR | Malaysian Ringgit | VUV | Vanuatu Vatu |
| FJD | Fiji Dollar | NZD | New Zealand Dollar | WST | Samoan Tala |
| FRF | French Franc | PGK | Papua New Guinean Kina | ZAR | South African Rand |
| GBP | United Kingdom Pound | SBD | Solomon Islander Dollar | | |

EmailAddress (input/output) Datatype: BSTR Max 255 bytes

The EmailAddress field can be used to store a customer's email address and will be returned in the transaction response. The response data along with the email address can then be used by the merchant to generate a notification/receipt email for the customer. This is an optional field.

EnableAddBillCard (input) Datatype: BSTR Max 1 character

The EnableAddbillCard field is used if subsequent billing is required. Setting this field to "1" will cause Payment Express to store the credit card details for future billing purposes. A DpsBillingId (Payment Express generated) or BillingId (merchant generated) is used to reference the card. More information on Token/Recurring billing can be found on [Token Billing Section](#).

MerchantReference (input/output) Datatype: BSTR Max 64 bytes

The Merchant Reference field is a free text field used to store a reference against a transaction. The merchant reference allows users to easily find and identify the transaction in Payline transaction query and Payment Express reports. The

merchant reference is returned in the transaction response, which can be used interpreted by the merchant website. Common uses for the merchant reference field are invoice and order numbers. This is an optional field.

ForcePaymentMethod (input) Datatype: String

Allows to force the specific payment method on the hosted payment page to use if the PxPayUserId has multiple payment methods configured. Only one of the below string values should be set. Please note certain payment methods such as Account2Account, Alipay, UPOP5 (China UnionPay) and WeChat, only accept the Purchase transaction type.

Any other values are deemed invalid according to the response.

If this field is not set then by default, multiple payment method options configured would be available on the hosted payment page.

- Account2Account
- Alipay
- AmexExpressWallet
- Card
- GECard
- MasterPass
- MonerisIOP
- Oxipay
- PayPal
- UPOP5
- UPOP
- VisaCheckout
- WeChat

PaymentMethod (output) Datatype: String

The final transaction response can contain the PaymentMethod field to indicate the specific payment method used for the transaction. If this field does not exist in the final transaction response then a card payment method was used by default.

ClientType (input) Datatype: BSTR

Allows the Client type to be passed through at transaction time. **Please note the feature may need to be enabled for the PxPay 2.0 user.** Recommended values for PxPay 2.0:

Internet = I

Recurring = R (if supported by your MID at your merchant bank expiry date will not be validated)

Please note the single letter input is case sensitive.

Please submit the full word or a single letter.

RecurringMode (input) Datatype: BSTR Max 255 bytes

In the RecurringMode request field, please set one of the card storage reason as the string listed below:

When **tokenising the card**:

- recurringinitial
- installmentinitial
- credentialonfileinitial
- unscheduledcredentialonfileinitial

When **rebilling the card** with a token returned on tokenising, set RecurringMode as one of:

- credentialonfile
- unscheduledcredentialonfile
- incremental
- installment
- recurring
- recurringnoexpiry
- resubmission
- reauthorisation
- delayedcharges
- noshow

Note: The RecurringMode string value should be set based on the merchant's business case or use case. Further details in [Token Billing section](#).

InstallmentNumber (input) Datatype: Number

Number based 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

The 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.

TxnData1, TxnData2, TxnData3 (input/output) Datatype: BSTR Max 255 bytes (each field)

The TxnData fields are free text fields that can be used to store information against a transaction. This can be used to store information such as customer name, address, phone number etc. This data is then returned in the transaction response and can also be retrieved from Payment Express reports. This is an optional field.

Note: storing large strings of information in the TxnData fields can cause the URI returned in the Request (Response) to exceed the max URL length for some browsers.

TxnType (input/output) Datatype: BSTR Max 8 characters

The TxnType can either be "Purchase" or "Auth".

A Purchase transaction type processes a financial transaction immediately and funds are transferred at next settlement cut-off. Basically a Purchase transaction debits the card in real time and if you need to cancel the transaction you will need to do a refund. An example of the Purchase transaction can be found on [Purchase Transaction Example](#) section.

The Auth transaction on the other hand, transfers no money. An Auth transaction type verifies that funds are available for the requested card and amount and reserves the specified amount. The amount is reserved for a period of up to 7 days (depending on merchant bank) before it clears. If you do not wish to take money from the customer you do nothing, and the reservation on the money will expire after 7 days. If you wish to process the payment a Complete transaction is sent to finalize the funds transfer.

For more information on the Auth and Complete transaction can be found on [Auth Transaction Example](#) section.

TxnId (input/output) Datatype: BSTR Max 16 bytes

On GenerateRequest XML it is a mandatory request field. Should always contain a unique, merchant application generated value that uniquely identifies the transaction. Used by Payment Express® to check for a duplicate transaction generated from Merchant web site. If a duplicate TxnId in the request is detected per PxPay account, the hosted payment page URI will not be generated and an error will be returned in the XML response of the GenerateRequest XML request and the merchant site should be informed of the result. Please ensure that the value is always unique per transaction request.

UriFail (input) Datatype: BSTR Max 255 bytes

Url of the page to redirect to if transaction failed. Please ensure the URL starts with the protocol, for example: "https:..." or "http:...". Query string characters are permitted. However percent-encoding the URL is not permitted. Please avoid using any custom ports for this URL.

UriSuccess (input) Datatype: BSTR Max 255 bytes

Url of the page to redirect to if transaction successful. Please ensure the URL starts with the protocol, for example: "https:..." or "http:...". Query string characters are permitted. However percent-encoding the URL is not permitted. Please avoid using any custom ports for this URL.

UriCallback (input) Datatype: BSTR Max 255 bytes

Externally accessible URL of page or resource to receive the notification on the conclusion of the HPP session when the transaction outcome is determined. Please ensure the URL starts with the protocol, for example: "https:..." or "http:...". Query string characters are permitted. Please avoid using any custom ports for this URL.

Opt (input) Datatype: BSTR Max 64 bytes

This optional parameter can be used to set a timeout value for the hosted payments page or block/allow specified card BIN ranges.

Timeout – A timeout (TO) can be set for the hosted payments page, after which the payment page will timeout and no longer allow a payment to be taken. Note: The default timeout of the created hosted payment page session is 72 hours. A specific timeout timestamp can also be specified via the request in Coordinated Universal Time (UTC). The value must be in the format "TO=yymmddHHmm" e.g. "TO=1010142221" for 2010 October 14th 10:21pm.

The merchant should submit the timeout value of when the payment page should timeout. One approach is to find the time (UTC) from when the GenerateRequest input XML document is generated and add on how much time you wish to give the customer before the payment page times out.

Example:

1. Customer finishes last stage of checkout process and clicks proceed to be redirected to the hosted payments page.
2. GenerateRequest Input XML document is generated with the required fields and the Opt field is populated with a timeout "TO=yymmddHHmm" that is 10 minutes later than the current time.
3. When the customer is redirected to the hosted payments page they will have 10 minutes to complete the payment before the page times out.

If the page times out the customer will be prompted with a "Declined – Transaction timed out" response, indicating that the page has timed out and payment was not processed. The customer will then be returned to the merchant's UrlFail page.

This feature is useful for merchants who wish control how much time a customer has to pay for the product e.g. concert tickets, where the merchant does not want people to be able to hold seats for an extended period of time due to limited availability.

Valid (output) Datatype: BSTR Max 1 character

The valid field indicates whether the initial request was valid or invalid i.e. "1" for valid and "0" for invalid.

URI (output) Datatype: BSTR

The URI field returns the URL including the encrypted transaction request that you will need to redirect the user to.

Response (input) Datatype: BSTR

The Response field should contain the encrypted URL response from Payment Express, which can be obtained from the "result" parameter in the URL string that is returned to your response page.

AmountSettlement (output) Datatype: BSTR Max 13 characters

The amount of funds settled with your bank for the transaction.

AuthCode (output) Datatype: BSTR Max 22 characters

Authorisation code returned for approved transactions from the acquirer.

CardName (output) Datatype: BSTR Max 16 bytes

The card type used for the transaction. Please refrain from processing any logic based on the value of CardName as it is subject to change from Payment Express. Please see examples of possible major card names below.

| CardName | CardNumber | CardType |
|------------|------------------|---------------------------------|
| Visa | 4111111111111111 | <CardName>Visa</CardName> |
| MasterCard | 5431111111111111 | <CardName>MasterCard</CardName> |
| Amex | 3711111111111114 | <CardName>Amex</CardName> |
| Diners | 360000000000008 | <CardName>Diners</CardName> |

CardNumber (output) Datatype: BSTR Max 16 bytes

The card number used for the transaction. The full credit card number isn't shown, however the bin range is given (first 6 characters) depending on masking pattern set.

DateExpiry (output) Datatype: BSTR Max 4 bytes

The expiry date of the card used in the transaction.

DpsTxnRef (input/output) Datatype: BSTR Max 16 bytes

A DpsTxnRef is returned for every transaction. If the transaction was approved, DpsTxnRef can be used as input to a Refund transaction i.e. used to specify a transaction for refund without supplying the original card number and expiry date.

Success (output) Datatype: Long

Indicates success or failure of the transaction. A value of 0 indicates the transaction was declined or there was an error. A value of 1 indicates the transaction was approved.

ResponseText (output) Datatype: BSTR Max 32 bytes

Response Text associated with the response code of the transaction.

DpsBillingId (input/output) Datatype: BSTR 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 is set to true indicating a token billing entry should be created. Information on Token Billing can be found on [Token Billing](#) section.

CardHolderName (output) Datatype: BSTR Max 64 bytes

The cardholder name as it appears on customer card.

CurrencySettlement (output) Datatype: BSTR Max 4 characters

Used to specify the currency that was used for the transaction: AUD, USD, NZD etc.

ClientInfo (output) Datatype: BSTR Max 15 characters

The IP address of the user who processed the transaction.

BillingId (input/output) Datatype: BSTR Max 32 characters

If a token based billing transaction is to be created, a BillingId may be supplied. This is an identifier supplied 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.

TxnMac (output) Datatype: BSTR Max 32 characters

Indication of the uniqueness of a card number.

CardNumber2 (output) Datatype: BSTR Max 16 characters

A token generated by Payment Express when adding a card for recurring billing. CardNumber2 is a 16 digit number which conforms to a Luhn 'mod 10' algorithm and has a 1-to-1 relationship with the actual card number used. Please contact Payment Express support if you would like to use this value.

Cvc2ResultCode (output) Datatype: BSTR 1 character

The result of CVC validation.

| Response Code | Definition | Interpreting Response Codes |
|---------------|--|---|
| 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 process. |

COMMON SCENARIOS

This section provides examples on the common PX Pay 2.0 scenarios.

All examples can be replicated on <https://demo.paymentexpress.com/SandboxMain.aspx> using our PX Pay 2.0 Sandbox tool (<https://demo.paymentexpress.com/Sandbox.aspx?endpoint=pxpay2SEC>).

PURCHASE TRANSACTION EXAMPLE

A Purchase transaction type processes a financial transaction immediately and funds are transferred at next settlement cut-off. Basically a Purchase transaction debits the card in real time and if you need to cancel the transaction you will need to do a refund.

Note additionally, if the PxPayUserId has multiple payment method configured by our activations team then you can force a specific payment method with the ForcePaymentMethod field as specified in the section [Element Descriptions](#). Please note certain payment methods only accept the Purchase transaction type.

Step 1 - GenerateRequest (Input XML Document): Post to Payment Express PX Pay 2.0 API (<https://sec.paymentexpress.com/pxaccess/pxpay.aspx>).

```
<GenerateRequest>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<MerchantReference>Purchase Example</MerchantReference>
<TxnType>Purchase</TxnType>
<AmountInput>1.00</AmountInput>
<CurrencyInput>NZD</CurrencyInput>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<EmailAddress>SampleUserId@paymentexpress.com</EmailAddress>
<TxnId>P03E57DA8A9DD700</TxnId>
<UrlSuccess>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlSuccess>
<UrlFail>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlFail>
</GenerateRequest>
```

Step 2 - Request (Output XML Document): Returned by Payment Express PX Pay 2.0 API for merchant to use to redirect customer to Payment Express Hosted Payments Page.

```
<Request valid="1">
<URI>https://sec.paymentexpress.com/pxmi3/EF4054F622D6C4C1B4F9AEA59DC91CAD3654CD60ED7ED04110CBC402959AC7
CF035878AEB85D87223</URI>
</Request>
```

Step 3 – Redirection to HPP: Customer is redirected to the Hosted Payments Page to enter their credit card details.

Page 1 – Hosted Payments Page (Enter CC details)

Page 2 - Transaction results

Note: Page 2 can be skipped so that redirection is done immediately after transaction is processed. Please refer to [HPP Customisation via Payline](#) for more information.

Step 4 – Redirection to UriSuccess/UriFail: Once transaction has been processed the customer is redirected back to the merchant's UriSuccess or UriFail depending on the transaction result. Transaction results are encrypted and returned in the "result" parameter of the URL string. You may also use the UriCallback to receive the notification on your server side.

```
https://demo.paymentexpress.com/SandboxSuccess.aspx?result=0000840000185416971799d38ef12345&userid=SampleUserId
```

Step 5 - ProcessResponse (Input XML Document): Generated by the merchant using PxPayUserId, PxPaykey and Response details to decrypt the transaction result. The Response is the URI obtained from the "result" parameter in the URL string that is returned to the UriSuccess/UriFail page in step 4.

```
<ProcessResponse>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<Response>00008400001853747f2bc6ded6012345</Response>
</ProcessResponse>
```

Step 6 - Response (Output XML Document): Transaction results decrypted and returned to merchant.

```
<Response valid="1">
<Success>1</Success>
<TxnType>Purchase</TxnType>
<CurrencyInput>NZD</CurrencyInput>
<MerchantReference>Purchase Example</MerchantReference>
<TxnData1></TxnData1>
<TxnData2></TxnData2>
<TxnData3></TxnData3>
<AuthCode>113837</AuthCode>
```

```

<CardName>Visa</CardName>
<CardHolderName>CARDHOLDER NAME</CardHolderName>
<CardNumber>411111.....11</CardNumber>
<DateExpiry>1111</DateExpiry>
<ClientInfo>192.168.1.111</ClientInfo>
<TxnId>P03E57DA8A9DD700</TxnId>
<EmailAddress></EmailAddress>
<DpsTxnRef>0000000604951234</DpsTxnRef>
<BillingId></BillingId>
<DpsBillingId></DpsBillingId>
<AmountSettlement>1.00</AmountSettlement>
<CurrencySettlement>NZD</CurrencySettlement>
<DateSettlement>20100924</DateSettlement>
<TxnMac>BDDDD619</TxnMac>
<ResponseText>APPROVED</ResponseText>
<CardNumber2></CardNumber2>
<Cvc2ResultCode>M</Cvc2ResultCode>
</Response>

```

AUTH TRANSACTION EXAMPLE

An Auth transaction type verifies that funds are available for the requested card and amount and reserves the specified amount. The amount is reserved for a period of up to 7 days (depending on merchant bank) before it clears. No funds are transferred. If you do not wish to take money from the customer you do nothing, and the reservation on the money will expire after 7 days. If you wish to process the payment a Complete transaction is sent to finalize the funds transfer. This is discussed on [Finalizing Auth Transactions](#) section.

Step 1 - GenerateRequest (Input XML Document): Post to Payment Express PX Pay 2.0 API (<https://sec.paymentexpress.com/pxaccess/pxpay.aspx>).

```

<GenerateRequest>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<MerchantReference>Auth Example</MerchantReference>
<TxnType>Auth</TxnType>
<AmountInput>1.00</AmountInput>
<CurrencyInput>NZD</CurrencyInput>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<EmailAddress>SampleUserld@paymentexpress.com</EmailAddress>
<TxnId>P03E890575E9D4A2</TxnId>
<UrlSuccess>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlSuccess>
<UrlFail>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlFail>
</GenerateRequest>

```

Step 2 - Request (Output XML Document): Returned by Payment Express PX Pay 2.0 API for merchant to use to redirect customer to Payment Express Hosted Payments Page.

```

<Request valid="1">
<URI>https://sec.paymentexpress.com/pxmi3/EF4054F622D6C4C1B4F9AEA59DC91CAD3654CD60ED7ED04110CBC402959AC7CF035878AEB85D87223</URI>
</Request>

```

Step 3 - Redirection to HPP: Customer is redirected to the Hosted Payments Page to enter their credit card details.

Page 1 – Hosted Payments Page (Enter CC details)

Page 2 - Transaction results

Note: Page 2 can be skipped so that redirection is done immediately after transaction is processed. The Amount shown on page 1 can also be hidden.

Step 4 – Redirection to UriSuccess/UriFail: Once transaction has been processed the customer is redirected back to the merchant's UriSuccess or UriFail depending on the transaction result. Transaction results are encrypted and returned in the "result" parameter of the URL string. You may also use the UriCallback to receive the notification on your server side.

```
https://demo.paymentexpress.com/SandboxSuccess.aspx?result=0000840000185416971799d38ef12345&userid= SampleUserId
```

Step 5 - ProcessResponse (Input XML Document): Generated by the merchant using PxPayUserId, PxPaykey and Response details to decrypt the transaction result. The Response is the URI obtained from the "result" parameter in the URL string that is returned to the UriSuccess/UriFail page in step 4.

```
<ProcessResponse>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<Response>00008400001853747f2bc6ded6012345</Response>
</ProcessResponse>
```

Step 6 - Response (Output XML Document): Transaction results decrypted and returned to merchant.

```
<Response valid="1">
<Success>1</Success>
<TxnType>Auth</TxnType>
<CurrencyInput>NZD</CurrencyInput>
<MerchantReference>Auth Example</MerchantReference>
<TxnData1>John Doe</TxnData1>
```

```
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<AuthCode>101828</AuthCode>
<CardName>Visa</CardName>
<CardHolderName>CARDHOLDER</CardHolderName>
<CardNumber>411111.....11</CardNumber>
<DateExpiry>1111</DateExpiry>
<ClientInfo>192.168.1.111</ClientInfo>
<TxnId>P03E890575E9D4A2</TxnId>
<EmailAddress>SampleUserld@paymentexpress.com</EmailAddress>
<DpsTxnRef>0000006049ac588</DpsTxnRef>
<BillingId></BillingId>
<DpsBillingId></DpsBillingId>
<AmountSettlement>1.00</AmountSettlement>
<CurrencySettlement>NZD</CurrencySettlement>
<DateSettlement>20100927</DateSettlement>
<TxnMac>BD43E619</TxnMac>
<ResponseText>APPROVED</ResponseText>
<CardNumber2></CardNumber2>
<Cvc2ResultCode>M</Cvc2ResultCode>
</Response>
```

FINALIZING AUTH TRANSACTIONS

A Complete transaction is sent at a later date to finalize the funds transfer. The merchant does not need to supply the card number or expiry date, only the DpsTxnRef returned from the original Auth transaction.

The amount sent in the Complete transaction can be less, more, or identical to the original amount specified in the original Auth transaction.

- If the Complete transaction is sent when the funds are still reserved (i.e. within the reservation period) and the amount is identical or less than the original Auth amount, then the funds are guaranteed to be available for the merchant to charge.
- If the amount is more than the original Auth amount then the amount up to the original Auth amount is guaranteed, however the additional amount is not.
- If the Complete transaction is sent when funds are not reserved (i.e. after the reservation period) the transaction can still be processed, however the funds are not guaranteed.
- One Complete transaction can be done per Auth transaction. If a Complete transaction is processed for a lesser amount than the original Auth amount, the remaining balance is no longer reserved.

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/services have not yet been delivered.

Please note that a complete transaction cannot be submitted via PX Pay 2.0, as the PX Pay 2.0 API is designed for the initial capture of card data only.

The Complete transactions will need to be processed manually using the Payline account (included with each PX Pay 2.0 account). Alternatively Payment Express can setup a Batch Processor, PX Post, or Web Services account for complete transactions only.

The following options are discussed in the next section.

- Option 1: PX Pay 2.0 > Payline
- Option 2: PX Pay 2.0 > Batch Upload
- Option 3: PX Pay 2.0 > Batch Processor
- Option 4: PX Pay 2.0 > PX Post
- Option 5: PX Pay 2.0 > Web Services

Option 1: PX Pay 2.0 > Payline

The Auth transaction can be finalized by processing a Complete transaction via Payline.

1. Browse to <https://sec.paymentexpress.com/pxmi3/logon>
2. Login using username and password.
3. Navigate to the Completions screen using the navigation menu on the left. Find the Auth transaction and click the "Complete" button.

The screenshot shows the Payment Manager interface with the following elements:

- Header:** paymentexpress logo, Brought to you by DPS, PAYMENT MANAGER, Timezone: NZT Host: PX1 via SECB
- Left Navigation Menu:** Billing, Setup Custom Features, Other, Reports, Transactions (Auth, Complete), Transaction Query, Change Password, Logoff
- Transaction Completion Search:** Start Date: 24 JUL 2013 04 12, End Date: 25 JUL 2013 04 12, Card Number, CardHolder Name, Merchant Reference, Search button
- Transaction Completion List:** Table with columns: Authorized Date, Card Number, Curr Amount, Type, DpsTxnRef. Row 1: 1, 20130724153227, 411111...11, 1.0000, A, 00000001088f0eeb

4. Enter the merchant reference and amount you wish to complete the transaction for, and click submit.

The screenshot shows the Payment Manager interface with the following elements:

- Header:** paymentexpress logo, Brought to you by DPS, PAYMENT MANAGER, Timezone: NZT Host: PX1 via SECB
- Left Navigation Menu:** Setup Custom Features, Other, Reports, Transactions (Auth, Complete), Transaction Query, Change Password, Logoff
- Completion:** Merchant Reference: Test1, Amount: 1.00 (NZD), Card Number: 411111...11, CardHolder Name: 123, Submit button

5. The transaction will be processed and the result will be returned. Clicking on the "Next Transaction" button will take you back to the Completions screen.

The screenshot shows the Payment Manager interface with the following elements:

- Header:** paymentexpress logo, Brought to you by DPS, PAYMENT MANAGER, Timezone: NZT Host: PX1 via SECB
- Left Navigation Menu:** Setup Custom Features, Other, Reports, Transactions (Auth, Complete), Transaction Query, Change Password, Logoff
- Authorization Result:** Date: 20130705045115, Response: APPROVED, ReCo: 00, Help Text, Authorization Code: 165115, Next Transaction button

Option 2: PX Pay 2.0 > Batch Upload

The Auth transaction can be finalized by processing a Complete transaction via the Payline Batch Processor.

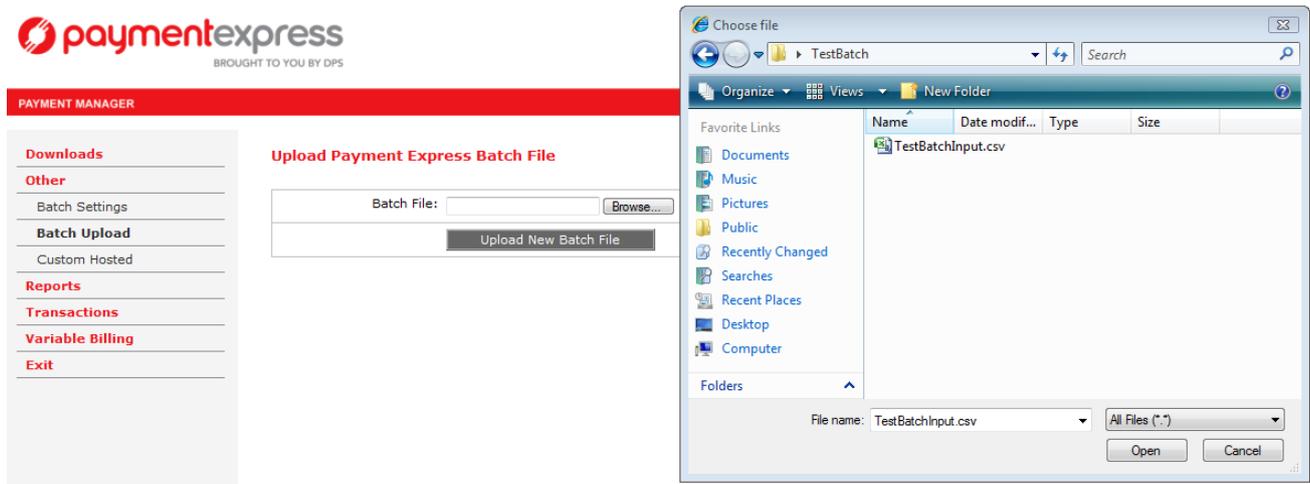
1. Process Auth transaction via PX Pay 2.0 and store the returned DpsTxnRef.
2. Create a batch input file according to specification. Technical documentation found here: http://www.paymentexpress.com/technical_resources/other_components/paylinebatch.html

Example batch input file contents:

C,9997,MerchantReference,,,1.00,0000006049ac588,,CardHolderName

The Complete transaction should include the DpsTxnRef returned from the original Auth. The DpsTxnRef in this example is "0000006049ac588".

3. Navigate to the Batch Upload screen using the navigation menu on the left. Click "Browse" to select your input file. Click "Upload New Batch File" to upload.



4. Once the batch file has been uploaded, the output file can be downloaded from the reports screen and transactions can be found in the transaction search screen.

Option 3: PX Pay 2.0 > Batch Processor

The Auth transaction can be finalized by processing a Complete transaction via the Payment Express® Batch Processor. The Batch Processor is designed to process input files containing credit card payment information for authorisation. Communication is done using HTTPS posts.

Technical documentation found here:

http://www.paymentexpress.com/technical_resources/other_components/batch_processor.html

1. Process Auth transaction via PX Pay 2.0 and store the returned DpsTxnRef.
2. Create a batch input file according to specification.

Example batch input file contents:

C,9997,MerchantReference,,,1.00,0000006049ac588,,CardHolderName

The Complete transaction should include the DpsTxnRef returned from the original Auth. The DpsTxnRef in this example is "0000006049ac588".

3. Drop the batch file into the "Input" folder to process the transaction.
4. Once the batch file has been uploaded, the output file will be generated and transactions can be found in the Payline transaction search screen.

Option 4-5: PX Pay 2.0 > PX Post/Webservice

The Auth transaction can be finalized by processing a Complete transaction via PX Post or Web Service.

PX Post: http://www.paymentexpress.com/technical_resources/ecommerce_nonhosted/pxpost.html#Authcomplete

Web Service: http://www.paymentexpress.com/technical_resources/ecommerce_nonhosted/webservice.html#authcomplete

1. Process Auth transaction via PX Pay 2.0 and store the returned DpsTxnRef.
2. Send a Complete transaction via PX Post or Web Service with the DpsTxnRef.

TOKEN BILLING

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 transaction is made. This functionality is implemented by providing the ability for a merchant to request Payment Express to capture and store a credit card number and expiry date and to link these stored details to a merchant supplied "BillingId" or Payment Express generated "DpsBillingId".

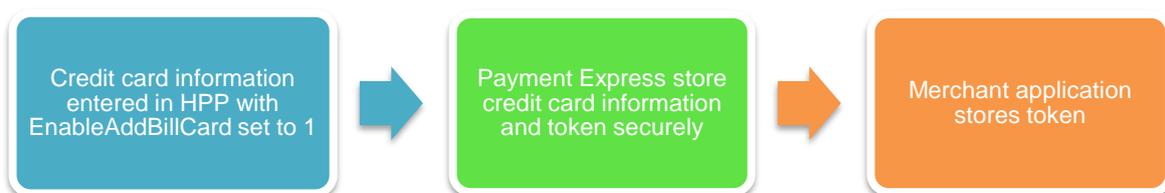
The **BillingId** is a 32 character field that contains a reference that is unique to the merchant's customer that will be associated with the credit card information stored securely at Payment Express. The BillingId is generated and supplied by merchant. This is undertaken during the Setup Phase.

The BillingId submitted does not have to be unique every time. If the BillingId of an existing card (stored at Payment Express) is submitted for a new card, the BillingId will only reference the new card moving forward. The old card can still be referenced by using the unique DpsBillingId.

The **DpsBillingId** is the same as the BillingId, but is generated by Payment Express not the merchant. A DpsBillingId will be generated for every transaction where the credit card information is to be stored. The returned value will be 16 characters in length and is unique. The merchant can choose to use the DpsBillingId or their own BillingId.

For subsequent charges to the card (Rebill Phase), the merchant does not need to supply the card number or expiry date, only the Token i.e. the BillingId or DpsBillingId originally associated during the Setup Phase.

SETUP PHASE



The setup phase consists of loading a card into Payment Express® with a transaction. The transaction can be a Purchase or Validate or Auth transaction type. The transaction can be an online \$0.00 or \$1.00 Validate which will only process a non-financial (zero dollar or no hold) transaction that is used check that the card and cardholder's account is valid. If the processing bank or acquirer does not support Validate then the Validate transaction request will be converted to an Auth transaction automatically.

Alternatively a \$1.00 Auth transaction type request will determine that the card is valid and not on hot or stolen card lists but depending on the processing bank or acquirer the transaction may incur a temporary financial hold of the transaction amount.

The Purchase transaction type is used if the card is to be charged with an amount and tokenised at the same time.

To add a card for future rebilling, send a PX Pay 2.0 transaction request including the following properties:

- EnableAddBillCard (Set to 1 when adding a card)
- RecurringMode (required)
- BillingId (optional)

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.

You can supply your own billing ID in the BillingId field or leave it blank and use the DpsBillingId returned in the response. We also recommend sending the unique TxnId and a MerchantReference. This will make searching and identifying tokens/cards much easier.

Token Creation Example

Step 1 - GenerateRequest (Input XML Document): Post to Payment Express PX Pay 2.0 API (<https://sec.paymentexpress.com/pxaccess/pxpay.aspx>).

```
<GenerateRequest>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<MerchantReference>Create Token Example</MerchantReference>
<TxnType>Validate</TxnType>
<AmountInput>1.00</AmountInput>
<CurrencyInput>NZD</CurrencyInput>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<EmailAddress>SampleUserId@paymentexpress.com</EmailAddress>
<TxnId>P03E9C7087908EB2</TxnId>
<EnableAddBillCard>1</EnableAddBillCard>
<RecurringMode>credentialonfileinitial</RecurringMode>
<UrlSuccess>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlSuccess>
<UrlFail>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlFail>
</GenerateRequest>
```

This transaction request has EnableAddBillCard set to 1. This will inform Payment Express to store credit card information for this request. Please note that we will be using DpsBillingId in this example. If you would rather use your own BillingId, simply add the BillingId field to the GenerateRequest and populate the value with the unique reference.

e.g. `<BillingId>abc12345xyz</BillingId>`

Also the RecurringMode is set to indicate the reason of card storage as a token. It should be set according to business case for the card tokenisation. The possible string values for RecurringMode field are listed above.

Step 2 - Request (Output XML Document): Returned by Payment Express PX Pay 2.0 API for merchant to use to redirect customer to Payment Express Hosted Payments Page.

```
<Request valid="1">
<URL>https://sec.paymentexpress.com/pxmi3/EF4054F622D6C4C1B4F9AEA59DC91CAD3654CD60ED7ED04110CBC402959AC7
CF035878AEB85D87223</URL>
</Request>
```

Step 3 - Redirection to HPP: Customer is redirected to the Hosted Payments Page to enter their credit card details.

Page 1 – Hosted Payments Page (Enter CC details)

Page 2 - Transaction results

Note: Page 2 can be skipped so that redirection is done immediately after transaction is processed. The Amount shown on page 1 can also be hidden.

Step 4 – Redirection to UriSuccess/UriFail: Once transaction has been processed the customer is redirected back to the merchant's UriSuccess or UriFail depending on the transaction result. Transaction results are encrypted and returned in the "result" parameter of the URL string. You may also use the UriCallback to receive the notification on your server side.

```
https://demo.paymentexpress.com/SandboxSuccess.aspx?result=0000840000185416971799d38ef12345&userid=SampleUserId
```

Step 5 - ProcessResponse (Input XML Document): Generated by the merchant using PxPayUserId, PxPaykey and Response details to decrypt the transaction result. The Response is the URI obtained from the "result" parameter in the URL string that is returned to the UriSuccess/UriFail page in step 4.

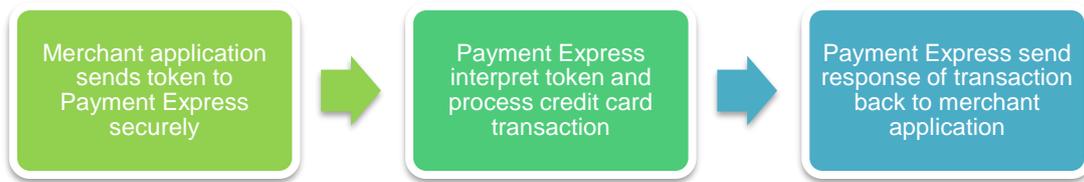
```
<ProcessResponse>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>17ce70a6025d8b925373066d3f71704a1868ea30a8d0485b6d086f722fd9997</PxPayKey>
<Response>00008400001853747f2bc6ded6012345</Response>
</ProcessResponse>
```

Step 6 - Response (Output XML Document)

```
<Response valid="1">
<Success>1</Success>
<TxnType>Validate</TxnType>
<CurrencyInput>NZD</CurrencyInput>
<MerchantReference>Create Token Example</MerchantReference>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<AuthCode>103426</AuthCode>
<CardName>Visa</CardName>
<CardHolderName>CREATE TOKEN EXAMPLE</CardHolderName>
<CardNumber>411111.....11</CardNumber>
<DateExpiry>1111</DateExpiry>
<ClientInfo>192.168.1.111</ClientInfo>
<TxnId>P03E9C7087908EB2</TxnId>
<EmailAddress>SampleUserld@paymentexpress.com</EmailAddress>
<DpsTxnRef>0000006049d1234</DpsTxnRef>
<BillingId></BillingId>
<DpsBillingId>000006000444444</DpsBillingId>
<AmountSettlement>1.00</AmountSettlement>
<CurrencySettlement>NZD</CurrencySettlement>
<DateSettlement>20100928</DateSettlement>
<TxnMac>BD43E619</TxnMac>
<ResponseText>APPROVED</ResponseText>
<CardNumber2></CardNumber2>
<Cvc2ResultCode>M</Cvc2ResultCode>
</Response>
```

- The unique DpsBillingId is generated by Payment Express and returned in this Response (Shown in the red text).
- The merchant can then store the DpsBillingId and use it for future billing purposes (Rebill phase).

REBILL PHASE



When the merchant wishes to bill a customer, the merchant application or Batch processor requests a new transaction and supplies the appropriate Token (BillingId or DpsBillingId), a MerchantReference, RecurringMode and the amount to be charged. The merchant does not need to supply the card number or expiry date, only the Token.

Payment Express® uses the Token and retrieves the credit card number and expiry date stored in the Setup Phase and processes a transaction to the associated card.

PxPay 2.0 can be used to rebill the token and billing can also be processed manually using the Payline account (included with each PX Pay 2.0 account). Alternatively Payment Express can setup a Batch Processor, PX Post or Web Service account for billing only.

The following options are discussed in the next section.

- Option 1: PX Pay 2.0 > PX Pay 2.0
- Option 2: PX Pay 2.0 > Batch Upload
- Option 3: PX Pay 2.0 > Batch Processor
- Option 4: PX Pay 2.0 > PX Post
- Option 5: PX Pay 2.0 > Web Services

Option 1: PX Pay 2.0

PX Pay 2.0 Rebilling Example

Step 1 - GenerateRequest (Input XML Document): Post to Payment Express PX Pay 2.0 API (<https://sec.paymentexpress.com/pxaccess/pxpay.aspx>).

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.

Example PxPay request:

```
<GenerateRequest>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890</PxPayKey>
<MerchantReference>Rebill Token Example</MerchantReference>
<TxnType>Purchase</TxnType>
<AmountInput>100.00</AmountInput>
<CurrencyInput>NZD</CurrencyInput>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<EmailAddress>SampleUserId@paymentexpress.com</EmailAddress>
<TxnId>P03E9C7087908EB2</TxnId>
<DpsBillingId>0000060004444444</DpsBillingId>
<RecurringMode>credentialonfile</RecurringMode>
<UrlSuccess>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlSuccess>
<UrlFail>https://demo.paymentexpress.com/SandboxSuccess.aspx</UrlFail>
</GenerateRequest>
```

This transaction request uses the DpsBillingID that was generated in the setup phase to recharge the card. Please note that we will be using DpsBillingId in this example. If you would rather use your own BillingId (that was generated in the setup phase), simply add the BillingId field to the GenerateRequest.

e.g. `<BillingId>abc12345xyz</BillingId>`

Step 2 - Request (Output XML Document): Returned by Payment Express PX Pay 2.0 API for merchant to use to redirect customer to Payment Express Hosted Payments Page.

```
<Request valid="1">
<URI>https://sec.paymentexpress.com/pxmi3/EF4054F622D6C4C1B4F9AEA59DC91CAD3654CD60ED7ED04110CBC402959AC7
CF035878AEB85D87223</URI>
</Request>
```

Step 3 - Redirection to HPP: Customer is redirected to the Hosted Payments Page to enter their Card Security Code.

Page 1 – Hosted Payments Page

(Card Number, Card Holder name and Expiry Date should be pre-defined)

Page 2 - Transaction results

Note: Page 2 can be skipped so that redirection is done immediately after transaction is processed. The Amount shown on page 1 can also be hidden. Refer to [Fail-Proof Result Notification](#) section.

Step 4 – Redirection to UriSuccess or UriFail: Once transaction has been processed the customer is redirected back to the merchant's UriSuccess or UriFail with the below GET message depending on the transaction result. Transaction results are encrypted and returned in the "result" parameter of the URL string. You may also use the UriCallback to receive the same notification on your server side.

```
https://demo.paymentexpress.com/SandboxSuccess.aspx?result=0000840000185416971799d38ef12345&userid=SampleUserId
```

Step 5 - ProcessResponse (Input XML Document): Generated by the merchant using PxPayUserId, PxPaykey and Response details to decrypt the transaction result. The Response is the URI obtained from the "result" parameter in the URL string that is returned to the UriSuccess/UriFail page in step 4.

```
<ProcessResponse>
<PxPayUserId>SampleUserId</PxPayUserId>
<PxPayKey>17ce70a6025d8b925373066d3f71704a1868ea30a8d0485b6d086f722fdd9997</PxPayKey>
<Response>00008400001853747f2bc6ded6012345</Response>
</ProcessResponse>
```

Step 6 - Response (Output XML Document)

```
<Response valid="1">
<Success>1</Success>
<TxnType>Purchase</TxnType>
<CurrencyInput>NZD</CurrencyInput>
<MerchantReference> Rebill Token Example</MerchantReference>
<TxnData1>John Doe</TxnData1>
<TxnData2>0211111111</TxnData2>
<TxnData3>98 Anzac Ave, Auckland 1010</TxnData3>
<AuthCode>103426</AuthCode>
<CardName>Visa</CardName>
<CardHolderName> REBILL TOKEN EXAMPLE</CardHolderName>
<CardNumber>411111.....11</CardNumber>
<DateExpiry>1111</DateExpiry>
<ClientInfo>192.168.1.111</ClientInfo>
<TxnId>P03E9C7087908EB2</TxnId>
<EmailAddress>SampleUserld@paymentexpress.com</EmailAddress>
<DpsTxnRef>0000006049d1234</DpsTxnRef>
<BillingId></BillingId>
<DpsBillingId>0000060004444444</DpsBillingId>
<AmountSettlement>1.00</AmountSettlement>
<CurrencySettlement>NZD</CurrencySettlement>
<DateSettlement>20100928</DateSettlement>
<TxnMac>BD43E619</TxnMac>
<ResponseText>APPROVED</ResponseText>
<CardNumber2></CardNumber2>
<Cvc2ResultCode>M</Cvc2ResultCode>
</Response>
```

The DpsBillingId will be the same to what was supplied by the merchant.

Option 2: PX Pay 2.0 > Batch Upload

Billing customers via Payline Batch Processor.

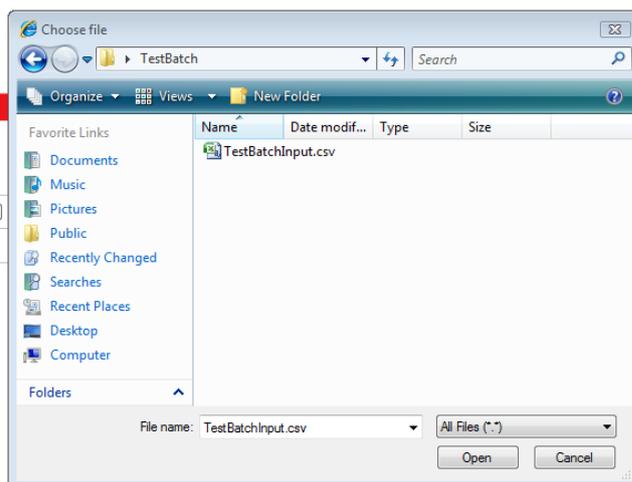
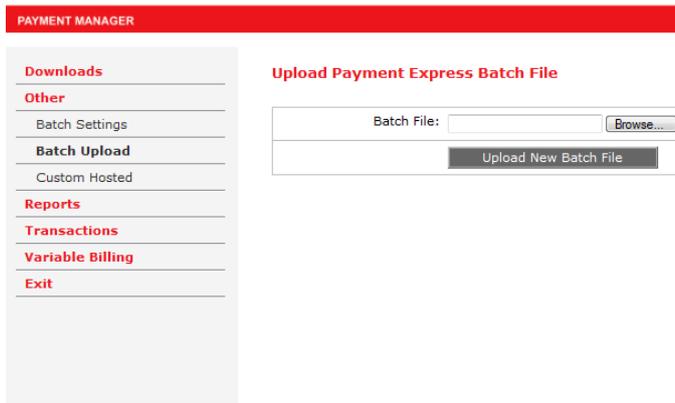
1. Merchant application will store the token returned in the PX Pay 2.0 Response.
2. Create a batch input file according to specification. Technical documentation found here: http://www.paymentexpress.com/technical_resources/other_components/paylinebatch.html

Example batch input file contents:

B,9997,MerchantReference,0000060004444444,,1.00,,CardHolderName

The billing transaction should include the token. The token (DpsBillingId) in this example is "0000060004444444".

3. Navigate to the Batch Upload screen using the navigation menu on the left. Click "Browse" to select your input file. Click "Upload New Batch File" to upload.



- Once the batch file has been uploaded, the output file can be downloaded from the reports screen and transactions can be found in the transaction search screen.

Option 3: PX Pay 2.0 > Batch Processor

Billing can be done via the Payment Express® Batch Processor. The Batch Processor is designed to process input files containing credit card payment information for authorisation. Communication is done using HTTPS posts.

Technical documentation found here:

http://www.paymentexpress.com/technical_resources/other_components/batch_processor.html

- Merchant application will store the token returned in the PX Pay 2.0 Response.
- Create a batch input file according to specification.

Example batch input file contents:

B,9997,MerchantReference,000060004444444,1.00,,CardHolderName

The billing transaction should include the token. The token (DpsBillingId) in this example is "000060004444444".

- Drop the batch file into the "Input" folder to process the transaction.
- Once the batch file has been uploaded, the output file will be generated and transactions can be found in the Payline transaction search screen.

Option 4-5: PX Pay 2.0 > PX Post or Webservice

- Merchant application will store the token returned in the PX Pay 2.0 Response.
- Send a Purchase transaction via PX Post or Web Service with the token.

Please visit one of the API options for rebilling or charging the card with a token:

- [Token Billing via PX Post](#)
- [Token Billing via Web Service](#)

UPOP (UNIONPAY ONLINE PAYMENTS)

PX Pay 2.0 is integrated with UPOP (UnionPay Online Payments). The integration is handled between Payment Express and UnionPay and the payment functionality is made available to all integrators of PX Pay 2.0.

There is no additional development work required to enable UPOP on existing PX Pay 2.0 integrations – we enable it as a Payment Method at the Payment Express end and the endpoint is the same as the standard PxPay 2.0 endpoint - however there are potential exceptions your application will need to handle. Because the UPOP payment method involves a redirect to a UnionPay hosted payment page for card holder card entry verification before redirection back to PxPay 2 with the transaction result, there is more potential for unusual navigation and unusual call back behaviour.

For this reason implementing PxPay 2.0's [FPRN](#) is essential. [FPRN](#) is of course enabled by default, but it requires development on your part to make use of the messages.

Additionally there is an exception scenario that has been noted with UPOP where the card holder after redirection from PxPay 2 to UPOP tries to click back in their browser to navigate back to PxPay 2. The transaction fails at this point but in some browsers a call back to your success / fail page may occur. If the result string from this call back is queried with a Process Response immediately then the Response will be as follows:

```
<Response valid="1">
  <Reco>N1</Reco>
  <ResponseText>No Transactions</ResponseText>
</Response>
```

Your application should be able to handle receiving this and to know that the transaction result can be ascertained later, either by waiting 10 minutes or so to re-query the result string, or by waiting to get the result via [FPRN](#). Such transactions are almost always declined transactions.

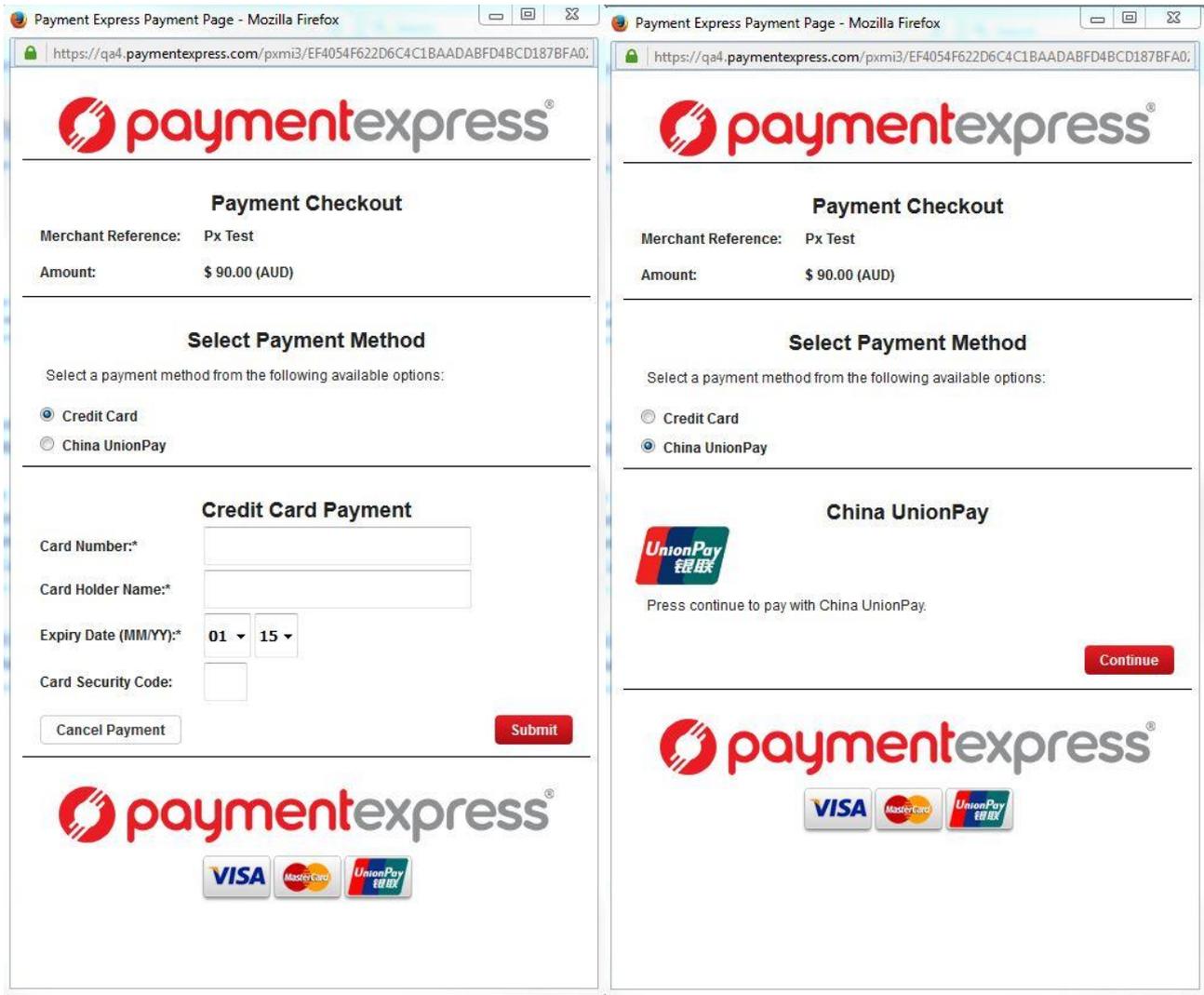
Please note: UPOP transactions do not support tokenisation.

Merchants can request to add UPOP to their PX Pay 2.0 account by contacting their Payment Express Account Manager to discuss on boarding process.

The following section shows the Cardholder journey via UPOP Authentication Interface.

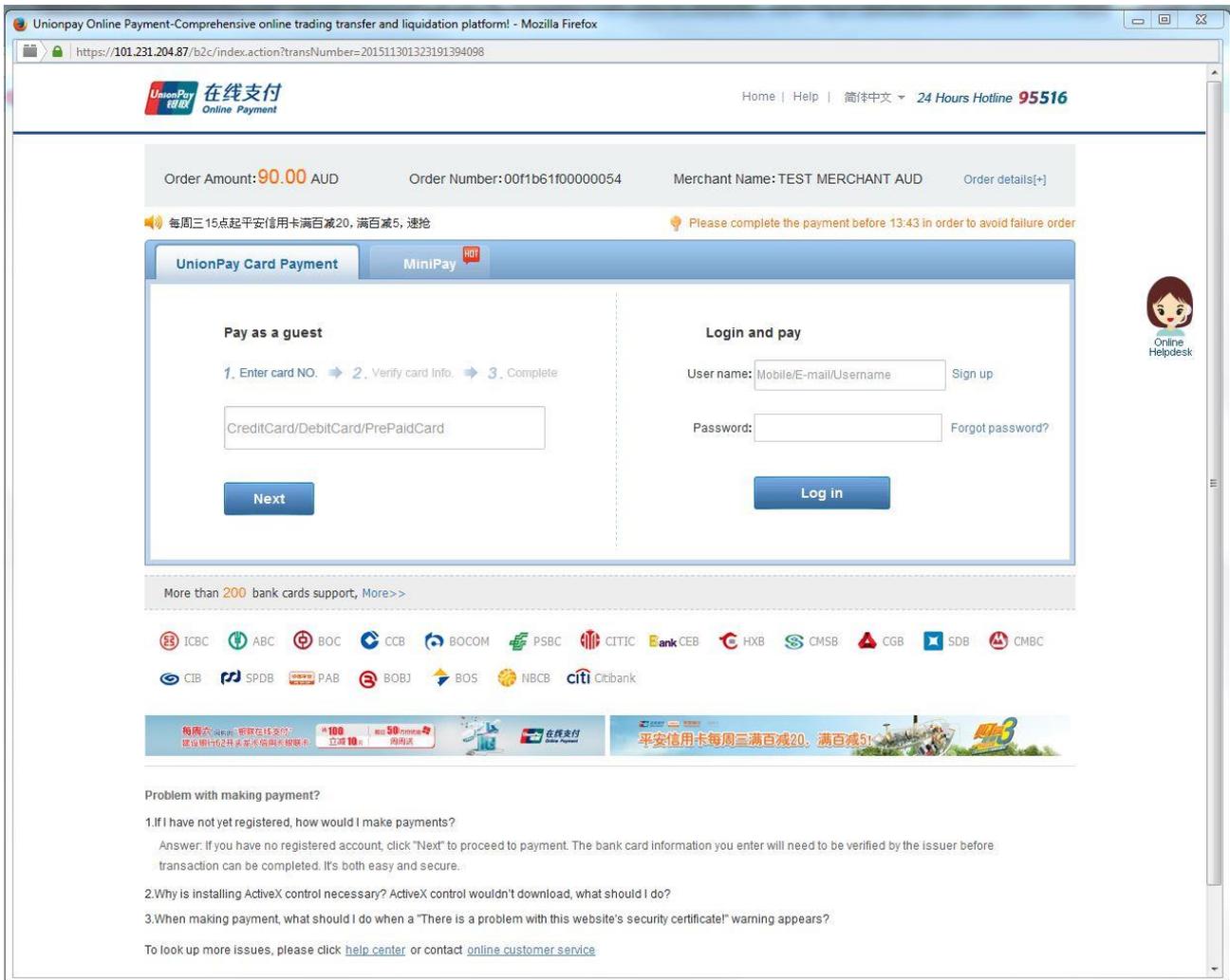
PX Pay 2.0 Select Payment Method: Once on the PX Pay 2.0 Hosted Payments Page, the cardholder is prompted with various Payment Methods options. Selecting the “China UnionPay” option will provide the cardholder the option to continue (redirect) to UPOP.

Please note: Credit Card payments can be completely segmented out from the UnionPay flow, depending on your requirements.



UPOP Authentication: Once on the UPOP page, the cardholder is prompted to login or pay as guest. These screens will be familiar to existing UPOP users.

The amount, order number reference and merchant names will be displayed at the top of the page. These details remain displayed throughout the process.



UPOP Payment: The cardholder can choose to login using their UPOP user/pass details OR pay as guest by entering their UnionPay card number.

The screenshot shows the UnionPay Online Payment interface in a Mozilla Firefox browser. The page title is "Unionpay Online Payment-Comprehensive online trading transfer and liquidation platform!". The URL is "https://101.231.204.87/b2c/index.action?transNumber=201511301323191394098".

At the top, there is a navigation bar with the UnionPay logo, "在线支付 Online Payment", and links for "Home | Help | 简体中文" and "24 Hours Hotline 95516".

The main content area displays transaction details: "Order Amount: 90.00 AUD", "Order Number: 00f1b61f00000054", and "Merchant Name: TEST MERCHANT AUD". There are promotional banners for "每周三15点起平安信用卡满百减20, 满百减5, 速抢" and "Please complete the payment before 13:43 in order to avoid failure order".

Below the banners, there are two main payment options: "UnionPay Card Payment" and "MiniPay". The "UnionPay Card Payment" option is selected and shows a "Pay as a guest" section with a three-step process: "1. Enter card NO.", "2. Verify card Info.", and "3. Complete". A card number "6250 XXXX XXXX 0014" is entered in a field, and a "Next" button is visible. The "Login and pay" section has fields for "User name: Mobile/E-mail/Username" and "Password", with "Sign up" and "Forgot password?" links, and a "Log in" button.

Below the payment options, there is a section for "More than 200 bank cards support, More>>" with logos for various banks including ICBC, ABC, BOC, CCB, BOCOM, PSBC, CITIC, Bank of China, HXB, CMSB, CGB, SDB, CMBC, CIB, SPDB, PAB, BOBJ, BOS, NBCB, and Citi.

At the bottom, there is a "Problem with making payment?" section with three questions and answers, and a link to "help center" or "online customer service".

UPOP Payment Security: The cardholder is prompted to enter various security details to complete the payment process. Security details such as CVC, SMS Code, and Web Verification Code. This is part of the standard UPOP security process so all existing UPOP users should be familiar.

Unionpay Online Payment-Comprehensive online trading transfer and liquidation platform! - Mozilla Firefox

https://101.231.204.87/b2c/foreignCard.action?transNumber=201511301323191394098

UnionPay 在线支付 Online Payment Home | Login | Register | Help | 简体中文 24 Hours Hotline 95516

Order Amount: 90.00 AUD Order Number: 00f1b61f00000054 Merchant Name: TEST MERCHANT AUD Order details[+]

每周三15点起平安信用卡满百减20, 满百减5, 速抢 Please complete the payment before 13.43 in order to avoid failure order

UnionPay Card Payment MiniPay

Security Control has been installed, your payment input is secured.

Card Number: (CreditCard) Choose other card

Expiration Date: MM YY
Please enter expiration date of your bank card. e.g. 09/13

CVN2:
Please enter the last three digits on the back of your card

Mobile Phone Number: 852-*****22 Is it changed?

SMS Code: Free SMS
Please enter the SMS verification code you received

Verification code:

I have read and accept [The User Agreement of UnionPay](#)

Confirm and Pay

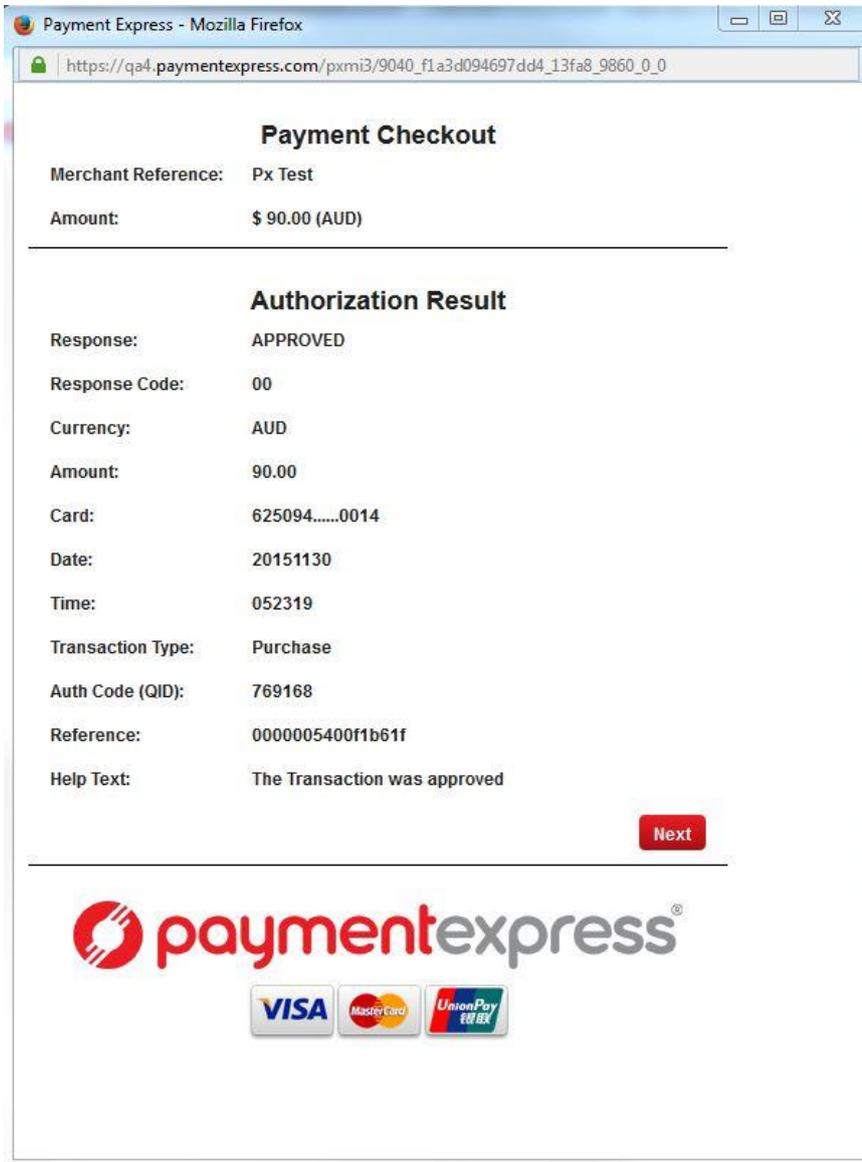
Problems with making payments?

- 1.What is a registered banking mobile number?
Answer: The registered banking mobile number is mobile number you registered with the bank while applying for bank card or processing bank card related business.
- 2.Why is installing ActiveX control necessary? ActiveX control wouldn't download, what should I do?
- 3.When making payment, what should I do when a "There is a problem with this website's security certificate" warning appears?

To look up more issues, please click [help center](#) or contact [online customer service](#)

PX Pay 2.0 Transaction Result: On successful completion of the UPOP payment process, the cardholder is redirected back to PX Pay 2.0 Hosted Payments Page and will be shown the transaction result details. The cardholder can then click the next button to be redirected back to the merchant website.

Please note: this step can be skipped if the merchant elects to display the results page themselves. This is configured on the PX Pay 2.0 user.



HPP CUSTOMISATION

Payment Express has provides a service which allows merchants to maintain a consistent look and feel between their website and the Hosted Payment Page. Merchants can select one of a number of pre-set styles and apply images and logos, colours, buttons and dynamically populated fields to the payment page. PxPay 2.0 also support merchant submitted custom CSS which provide a greater flexibility in interface design.

- Ability to customise the look and feel of the hosted payments page.
- Pre-set style sheets available to customise the look and feel of the hosted payments page.
- Support for merchant submitted custom header logo and background images.
- Support for merchant submitted custom CSS.
- Display logos of accepted cards.
- Display or hide Payment Checkout information such as Amount, Merchant Reference and TxnData1-3 fields.
- Ability to customise all field labels & titles.
- Preview option available before deployment.
- Display or Skip Payment Result page.

Requirements

- A PxPay 2.0 development or live account.
- “Setup Custom Feature” on your Payment Manager logon at <https://sec.paymentexpress.com/pxmi3/logon>

(Please contact sales@paymentexpress.com if you do not have a Payline account and/or the Custom hosted screen available on your Payline portal.)

HPP CUSTOMISATION VIA PAYLINE

Payment Express provides a service which allows merchants to maintain a consistent look and feel between their website and the Hosted Payment Page. Merchants can select one of a number of pre-set styles and apply images and logos, colours, buttons and dynamically populated fields to the payment page.

➤ Your Setup Custom Features screen will look like this:

PxPay Custom Hosted

| Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Preview Page |
|-------------------------|--------------------------|----------------------------|----------------------|--------|------------|--------------|--------------|
| Label Title: | <input type="text"/> | Label Details Description: | <input type="text"/> | | | | |
| Hide Amount | <input type="checkbox"/> | Label Amount: | <input type="text"/> | | | | |
| Hide TxnData1 | <input type="checkbox"/> | Label TxnData1: | <input type="text"/> | | | | |
| Hide TxnData2 | <input type="checkbox"/> | Label TxnData2: | <input type="text"/> | | | | |
| Hide TxnData3 | <input type="checkbox"/> | Label TxnData3: | <input type="text"/> | | | | |
| Hide Merchant Reference | <input type="checkbox"/> | Label Merchant Reference: | <input type="text"/> | | | | |
| Hide Currency | <input type="checkbox"/> | | | | | | |

For the above fields you can:

- Hide/unhide the label fields (please use these settings rather than hide/unhide via Custom CSS)
- Set a customised label for the fields

In the Payment Page tab, you can configure various settings related to the hosted payment page with the payment methods including fields related to the card payment form.

In the Result Page tab, if you wish to skip the second hosted page which displays the final transaction results, “**Disable Result Page (SkipPage2)**” can be used.

➤ You will be able change the interface styles on the Page Style tag. There are 1-6 legacy and newer 7-13 styles available.

PxPay Custom Hosted

| Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Preview Page |
|--------------------------|--------------|---------------------------|--------------------------|--------|------------|--------------|--------------|
| Primary Css: | STYLE1 | Disable Browser Based Css | <input type="checkbox"/> | | | | |
| Background Colour (Hex): | STYLE1 | Panel Colour (Hex): | <input type="text"/> | | | | |
| Title Text Colour (Hex): | STYLE3 | Label Text Colour (Hex): | <input type="text"/> | | | | |
| Value Text Colour (Hex): | STYLE5 | Button Colour (Hex): | <input type="text"/> | | | | |
| InputBox Colour (Hex): | STYLE6 | Link Colour (Hex): | <input type="text"/> | | | | |

➤ You will be able to preview your styles from the preview page.

PxPay Custom Hosted

| Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Preview Page |
|--|--------------|-------------|--------------|--------|------------|--------------|--------------|
| <div style="background-color: red; color: white; padding: 5px; display: inline-block;">Preview (you must update any changes first)</div> | | | | | | | |

HPP Page 1 Sample - Customisable Fields (Payment Details & Payment Page)

Images

- Dps Header Logo
- Background Logo
- Customer Logo

Payment Details

- Label Title
- Label Amount
- Label TxnData1
- Label TxnData2
- Label TxnData3
- Label Merchant Reference
- Hide Currency
- Hide Amount
- Hide TxnData1
- Hide TxnData2
- Hide TxnData3
- Hide Merchant Reference

Images

- Dps Footer Logo

Payment Page

- Label CardNumber
- Label NameOnCard
- Label Payment Method
- Label Submit Button
- Label ExpiryDate
- Label CardSecurityCode
- Label IssueNumber
- Label Cancel Button
- Hide Cancel Button
- Show IssueNumber
- Show MasterCard Logo
- Show MasterCard SecureCode Logo
- Show JCB Logo
- Show EzibuyGift Logo
- Show Visa Logo
- Show VerifiedByVisa Logo
- Show Amex SafeKey Logo
- Show Diners Logo
- Show Cvc2 Help Link
- Minimum Card Digits
- Maximum Card Digits

PxPay Custom Hosted

| Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Cardholder Receipt | Preview Page |
|---------------------------------|--------------------------|-------------|--------------|------------------------------------|--------------------|--------------|--------------------|--------------|
| Label Title: | Payment Checkout | | | Label Details Description: | | | | |
| Hide Amount | <input type="checkbox"/> | | | Label Amount: | Amount | | | |
| Hide TxnData1 | <input type="checkbox"/> | | | Label TxnData1: | Order Line 1 | | | |
| Hide TxnData2 | <input type="checkbox"/> | | | Label TxnData2: | Order Line 2 | | | |
| Hide TxnData3 | <input type="checkbox"/> | | | Label TxnData3: | Order Line 3 | | | |
| Hide Merchant Reference | <input type="checkbox"/> | | | Label Merchant Reference: | Merchant Reference | | | |
| Label Merchant Authorized Name: | Merchant Name | | | Label Merchant Authorized Address: | Merchant Address | | | |
| Hide Currency | <input type="checkbox"/> | | | | | | | |

✓ Apply
↺ Reset

HPP Page 1 Sample - Customisable Fields (Page Style)

Payment Checkout

Merchant Reference: **Sample Merchant Reference**

Order Line 1: **Sample TxnData1**

Order Line 2: **Sample TxnData2**

Order Line 3: **Sample TxnData3**

Amount: **1.00 (DFLT)**

Select Payment Method

Select a payment method from the following available options:

Credit Card

Pago

Credit Card Payment

Card Number:

CardHolder Name:

Expiry Date (MM/YY):

Card Security Code:

Submit

Cancel Payment



PxPay Custom Hosted

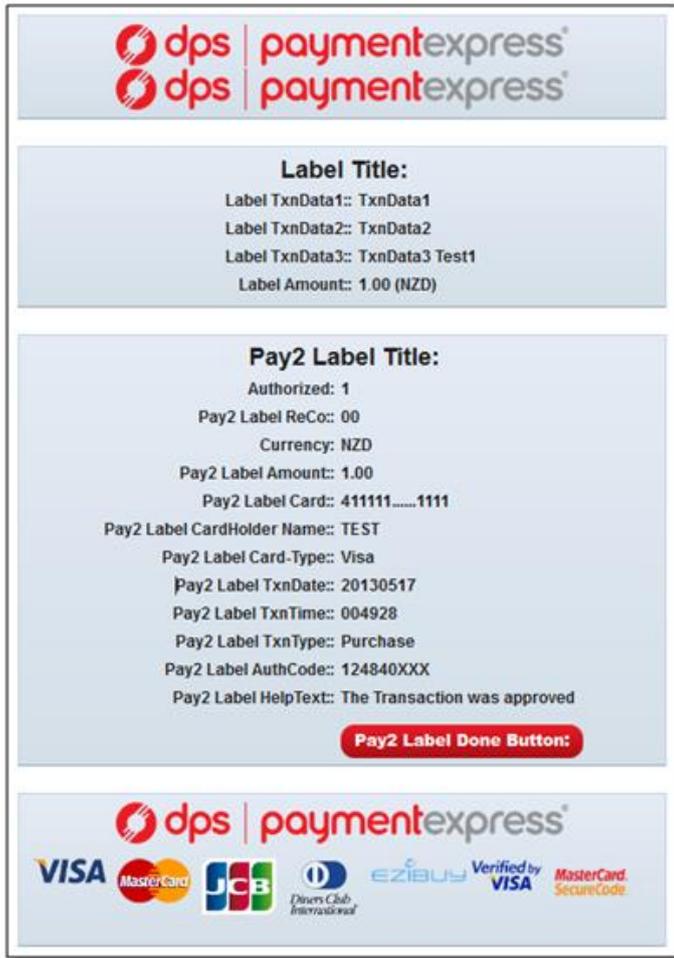
Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Cardholder Receipt | Preview Page

| | | | |
|----------------------------|----------------------|---------------------------|--------------------------|
| Primary Css: | STYLE8 | Disable Browser Based Css | <input type="checkbox"/> |
| Background Colour (Hex): | <input type="text"/> | Panel Colour (Hex): | <input type="text"/> |
| Panel Border Colour (Hex): | <input type="text"/> | Title Text Colour (Hex): | <input type="text"/> |
| Label Text Colour (Hex): | <input type="text"/> | Value Text Colour (Hex): | <input type="text"/> |
| Button Colour (Hex): | <input type="text"/> | InputBox Colour (Hex): | <input type="text"/> |
| Link Colour (Hex): | <input type="text"/> | | |

✓ Apply

↺ Reset

HPP Page 2 Sample - Customisable Fields (Result Page)



| Result Page |
|-----------------------|
| Label Title |
| Label Amount |
| Label Card |
| Label CardHolder Name |
| Label Card-Type |
| Label TxnDate |
| Label TxnTime |
| Label TxnType |
| Label ReCo |
| Label ResponseText |
| Label HelpText |
| Label AuthCode |
| Label Done Button |
| Label Cancel Button |
| Label Retry Button |
| Hide Currency |
| Hide Amount |
| Hide Card |
| Hide Card Holder Name |
| Hide Card-Type |
| Hide TxnDate |
| Hide TxnTime |
| Hide TxnType |
| Hide ReCo |
| Hide ResponseText |
| Hide HelpText |
| Hide AuthCode |

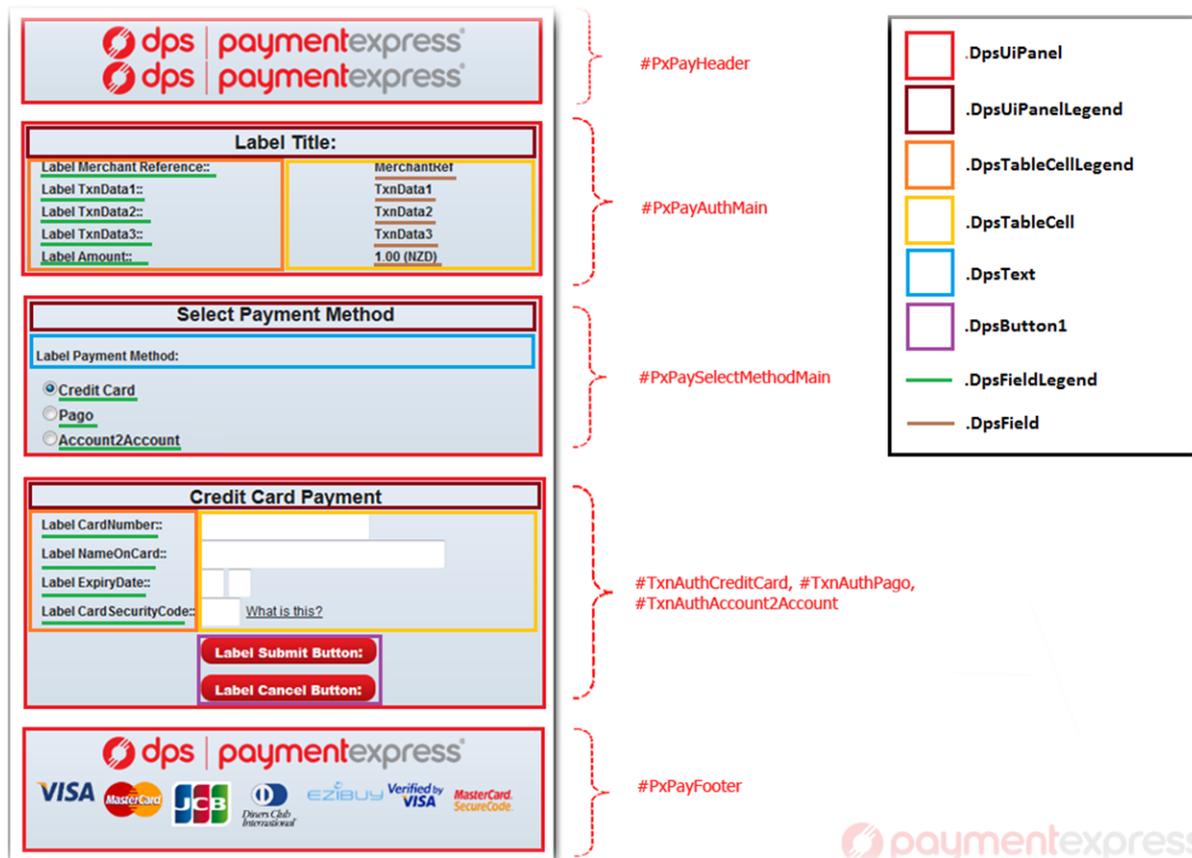
PxPay Custom Hosted

| Payment Details | Payment Page | Result Page | Payform Page | Images | Page Style | A2A Specific | Cardholder Receipt | Preview Page |
|---|--------------|--------------------------|-----------------------------|--------|------------|--|--------------------|--------------------------|
| Disable Result Page (SkipPage2) | | <input type="checkbox"/> | | | | Label Title: | | Authorization Result |
| Hide Currency | | <input type="checkbox"/> | | | | Label Currency: | | Currency |
| Hide Amount | | <input type="checkbox"/> | | | | Label Amount: | | Amount |
| Hide Card | | <input type="checkbox"/> | | | | Label Card: | | Card |
| Hide Card Holder Name | | <input type="checkbox"/> | | | | Label CardHolder Name: | | Card Holder |
| Hide Card-Type | | <input type="checkbox"/> | | | | Label Card-Type: | | Card Type |
| Hide TxnDate | | <input type="checkbox"/> | | | | Label TxnDate: | | Date |
| Hide TxnTime | | <input type="checkbox"/> | | | | Label TxnTime: | | Time |
| Hide Txn Datetime (User TZone) | | <input type="checkbox"/> | | | | Label Txn Datetime (User TZone): | | Transaction Date |
| Hide TxnType | | <input type="checkbox"/> | | | | Label TxnType: | | Transaction Type |
| Hide ReCo | | <input type="checkbox"/> | | | | Label ReCo: | | Response Code |
| Hide ResponseText | | <input type="checkbox"/> | | | | Label ResponseText: | | Response |
| Hide HelpText | | <input type="checkbox"/> | | | | Label HelpText: | | Help Text |
| Hide AuthCode | | <input type="checkbox"/> | | | | Label AuthCode: | | Auth Code |
| Hide DpsTxnRef | | <input type="checkbox"/> | | | | | | |
| Label Payment Result Mail Hasbeen Sent: | | <input type="checkbox"/> | The transaction result mail | | | Label Send Payment Result to Cardholder: | | Send me an email receipt |
| Label Email: | | <input type="text"/> | Email | | | Label Done Button: | | Next |
| Label Retry Button: | | <input type="text"/> | Retry | | | Label Cancel Button: | | Cancel Payment |
| Label Result Description: | | <input type="text"/> | | | | | | |

MERCHANT SUBMITTED CSS

The merchant can also modify the CSS code from the default CSS template and customise the payment page interface to suit their purpose. For sample CSS files, please see our website.

The following diagram outlined the most common Class (.) and ID (#) selectors that can be adjusted using CSS.



Please ensure the CSS file ONLY contains CSS code. No HTML and Java Script is allowed in the file.

Please use the custom hosted settings to hide some elements in the payment page such as label fields or buttons. Please avoid hiding elements from the CSS itself.

On the preview page, you can simulate a test transaction by using test credit card credentials (credit card number: 4111111111111111, using any expiry date in the future. To update changes you have made click 'Update'.

Upload a merchant logo or background images:

Please email your images to support@paymentexpress.com quoting your PXPAY ID in the email. Please send image files in web format only (.jpg, .jpeg, .gif, .png). Please ensure image file is less than 40KB in file size.

Submit a customized CSS file:

Please email your CSS file to support@paymentexpress.com quoting your PXPAY ID in the email. You can minify CSS.

Please ensure the CSS file is less than 48KB and contains CSS code only. No custom HTML and Javascript are allowed.

Apply Changes and bring your changes live

To activate the customised hosted payments page that you have just created (bring the changes live) please email support@paymentexpress.com quoting your PXPAY ID in the email.

FAIL-PROOF RESULT NOTIFICATION (FPRN)

Fail Proof Result Notification (FPRN) is a server side service that provides additional assurance that the merchant website will receive notification regarding the outcome of transactions completed via the Payment Express Hosted Payment Page (HPP).

FPRN helps cater for the possibility that a user may not successfully navigate to the nominated success or failure URL enabling the merchant web application to acknowledge the outcome of the transaction. The user could close their browser or otherwise navigate away from the Payment Express hosted payment page once they have been informed of the transaction outcome. The merchant's web server may be temporarily unavailable as the transaction is completed and therefore unable to recognise that a transaction has taken place. Using the FPRN service the merchant website is virtually guaranteed to receive notification of the each and every transaction.

FPRN is highly recommended by Payment Express and is enabled on all new accounts by default. The service ensures that the following processes occur for every transaction performed via hosted payment page:

1. Once the transaction outcome is determined for the HPP session, a background process at Payment Express makes an **HTTP GET request** to the merchant nominated URL set on the GenerateRequest fields: **UrlSuccess, UrlFail or UrlCallback**. The URLs should resolve to a HTTP 200, 302 or 303 response externally.
2. If the merchant nominated URL is unreachable or returns any HTTP status code other than 200 or 404 the **notification (HTTP GET) is retried** up to a maximum of six times.
For example, a 500 HTTP status code, indicating a temporary problem at the client site, will cause a retry.
3. Our FPRN service will stop sending any notifications immediately on receiving a 404 (Not Found) or 502 (Bad Gateway) HTTP status code.

We can configure the FPRN HTTP status code behaviour further per API user if required. Also per API user we can configure additional query string parameter key value pair for the FPRN GET requests only for the specific API user account. This may help lock a static authentication value for server side notifications. Please contact support@paymentexpress.com to get assistance with this. Otherwise we recommend to simply set the UrlCallback field per GenerateRequest.

In order to ensure that the web application is in the best position to acknowledge the outcome of each and every transaction certain guidelines should be followed.

The merchant web application should not:

- Filter or base any conditional logic upon the originating IP address as this can vary.
- Depend upon receiving one and only one HTTP GET request for the success, fail or callback URL from the Payment Express FPRN system (multiple requests may be sent).

The merchant web application should:

- Rely on the UrlSuccess, UrlFail or UrlCallback field to receive the notification GET message. Note: UrlCallback can be used to receive the server side notification (regardless of transaction outcome) separately to the same GET redirects coming from a client side such as a web browser.
- Get the 'result' query string parameter value from the GET notification request to UrlSuccess, UrlFail or UrlCallback. Your URL will have that appended 'result' query string parameter containing the encrypted key for the transaction details. The query string also contains the 'userid' to identify the API user associated to the transaction outcome.
- Use the userid as PxPayUserId, include the PxPayKey stored securely in your server side and result parameter as Response value for the final ProcessResponse request to get the final decrypted transaction outcome details.
- Determine if a database operation or some form of communication such as generating an order record or sending an email is required. Generally this will mean that the application needs to be aware if these actions have been taken previously for the particular transaction or not. TxnId should be used for this purpose.

Note: The URL at which the merchant website will process FPRN requests must be exposed via standard internet ports i.e. port 80 or port 443 for SSL/TLS traffic. When specifying UrlSuccess, UrlFail or UrlCallback field values do not specify a non-standard port number within the URL.

Please note FPRN is turned on by default.

3D SECURE

As part of Payment Express' continuing efforts to promote best practice solutions, the Hosted Payments Package allows for merchants to participate in the Verified by Visa and MasterCard's SecureCode schemes at no additional cost.

These latest schemes allow merchants to enjoy:

- Fraud reduction
- Reduced liability
- Reduced operational costs
- Increase sales and marketing opportunities



These schemes allow merchants' to prove to their potential customers that they are serious about protecting the identity and integrity of their customers' sensitive card details. In doing so, the merchant can potentially increase sales generated through their web services, by converting 'window-shoppers' to 'shoppers'.

These schemes require the card holder to enter a password (only if the card holder is enrolled) as an additional step to entering their credit card number and expiry date that is verified by the issuing bank. Once the issuing bank has verified the password, the user is then redirected back to the merchant's web site.



Payment Express have implemented a MPI (Merchant Plug-In) that securely allows for these schemes to be implemented with no development cost or hassle to the merchant whatsoever.

The MPI is compliant with the latest 3D secure technology and Visa estimates that with these new security mechanisms in place, merchants will experience an increase in sales generated and lower operational costs – having to deal with fewer charge backs.

TROUBLESHOOTING & FAQs

I have set the response pages URLs in my transaction request form, but I'm not getting redirected back to the pages I asked for.

You need to use a fully qualified address for example <http://www.yourwebsite.com/response.xxx>. You cannot just use response xxx etc.

How do I get my logo to show on the Payments Page?

To upload a merchant logo to your payments page, please email support@paymentexpress.com quoting your PX Pay 2 Id in the email.

What happens if the customer doesn't click next on the payment page and closes their window, how will the response message get back to my success/failure pages?

There is a background post as well that sends the response to your site that you requested and the customer has also seen the response by that time. Clicking the next button to return to your site will also send a redundant post.

Why do I get an error page stating that the 'Root element is missing'?

When the URL is provided as part of an XML document certain characters such as the ampersand must be escaped in order to maintain the integrity of the document. Before directing the browser to the URL provided the escaped characters should be unescaped.

Why can't Internet Explorer display my success or failure pages?

Internet Explorer limits URLs to 2083 characters, above which it will display a page stating that 'Internet Explorer cannot display the webpage'. The lengths of the query strings used on the payment pages and of those appended to the success and failure URLs can sometimes cause the URLs to exceed the maximum allowable in Internet Explorer. This will generally only occur when large amounts of transaction data are being passed to Payment Express and are subsequently encrypted and become part of the query string. Please contact the Payment Express support team to ask about URL compression or reduce the amount of transaction data being passed in optional fields.

I am unable to process Japanese currency, every time I submit the transaction request I receive the error code "IU". I have no problems processing in NZD and AUD.

When submitting transactions for currencies with no decimal division of units such as the Japanese Yen (JPY), the AmountInput must be in an appropriate format e.g. "1000" for 1000 Yen. If "1000.00" is submitted in AmountInput with CurrencyInput set to JPY, an error will occur and response code "IU" will be returned.

Can I process refunds via the Hosted Payments Page?

No, the HPP solution is designed for initial capture of card data only. Refunds can be done via Payline or our merchant hosted API solutions (PX Post, Web Service, DPSSAuth SSL).

REQUEST RESPONSE CODES

During development if the URI is not received after a transaction request you may receive a Request XML document (response data) with children elements Reco and ResponseText from Payment Express. The Reco (Response code) and ResponseText provides an indication of the error (see an **example of the XML below**). Common error codes and the potential cause for the error are listed below. Please note a Reco may be received if the integration is not receiving the hosted payment page URI. To debug a possible request issue, please ensure your integration is logging the response data after a [GenerateRequest](#) is sent to Payment Express.

Please note these error codes are related to the functionality of PxPay 2.0 interface itself and are different to the ReCo related to the host, bank or acquirer that you receive in the final transaction [response XML document](#).

```
<Request valid="1">
<Reco>IP</Reco>
<ResponseText>Invalid Access Info</ResponseText>
</Request>
```

Please note that error codes are subject to change and under no circumstances should you base any application logic on error codes returned. If you cannot identify the issue from the list above please contact support@paymentexpress.com

| Code | Description |
|------|--|
| IC | Invalid Key or Username. Also check that if a TxnId is being supplied that it is unique. |
| ID | Invalid transaction type. Ensure that the transaction type is either Auth or Purchase. |
| IK | Invalid UrlSuccess. Ensure that the URL being supplied does not contain a query string. |
| IL | Invalid UrlFail. Ensure that the URL being supplied does not contain a query string. |
| IM | Invalid PxPayUserId. |
| IN | Blank PxPayUserId. |
| IP | Invalid Access Info. Ensure the PxPayID and/or key are valid. |
| IQ | Invalid TxnType. Ensure the transaction type being submitted is either "Auth" or "Purchase". |
| IT | Invalid currency. Ensure that the CurrencyInput is correct and in the correct format e.g. "USD". |
| IU | Invalid AmountInput. Ensure that the amount is in the correct format e.g. "20.00". |
| NF | Invalid Username. |
| NK | Request not found. Check the key and the mcrypt library if in use. |
| NL | User not enabled. Contact Payment Express. |
| NM | User not enabled. Contact Payment Express. |
| NN | Invalid MAC. |
| NO | Request contains non ASCII characters. |
| NP | Closing Request tag not found. |
| NQ | User not enabled for PxPay 2.0. Contact Payment Express. |
| NT | Key is not 64 characters. |

GO LIVE

When testing is done and you are ready to start accepting live payments we will link your Payment Express account to your bank/acquirer merchant account(s).

In order for Payment Express to process live payments, please complete the following steps:

1. Register for the Payment Express software at <https://sec.paymentexpress.com/pxmi3/apply>
2. Once applied; a Payment Express consultant will make contact with you and explain the setup procedure and put you in contact with your chosen bank.
3. You will need to apply for an "ecommerce merchant account" with your bank and specify that payments will be processed through "Payment Express".
4. Once you have completed the process with the bank, they will issue you with a merchant ID / Merchant number. Payment Express will need this in order to activate your account for live payments.
5. Once Payment Express have been provided with the Merchant ID and confirmation that your website is ready to go live; we can activate your payment gateway with 3 working days' notice.

If unsure about any step above, please contact Payment Express on the details provided below for your region:
http://www.paymentexpress.com/contact_us.html