



# Payment Express<sup>®</sup> Ecommerce

SOAP Web Service

## CONTENTS

OVERVIEW .....	2
SOAP OPERATIONS .....	3
SubmitTransaction / SubmitTransaction2 .....	3
UpdateCard.....	8
EXCEPTION HANDLING .....	9
GetStatus / GetStatus2 .....	9
REFUNDS.....	12
AUTH-COMPLETE.....	12
Overview.....	12
Authorization.....	12
Complete .....	12
TOKEN BILLING .....	13
Overview.....	13
Setup Phase .....	14
Rebill Phase.....	14
Recurring Transactions .....	14
EXTENDED AIRLINE BOOKING DATA.....	15
TEST CARDS.....	15
PROPERTIES DESCRIPTION.....	16

## OVERVIEW

The SOAP Web Service (WS) Client Application connects directly to the Payment Express Host via HTTPS POSTs.

Client applications communicate with the web service using SOAP (Simple Object Access Protocol). SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

The web service live endpoint address is <https://sec.paymentexpress.com/WS/PXWS.asmx>. Each method call is self documented at this address by clicking on each method.

The Web Service Schema (WSDL) can be found at <https://sec.paymentexpress.com/WS/PXWS.asmx?WSDL>

*Note: test endpoint address is <https://qa2.paymentexpress.com/WS/PXWS.asmx>.*

### Lax Versioning

New features and response elements may be added to the Web Service over time. Any future updates will be carried out under the understanding that lax versioning is implemented on all clients consuming it. This prevents merchants needing to connect to a new endpoint in order to take advantage of any new features, while also ensuring that any new additions do not break the merchant's existing application.

For these reasons, please ensure that your web service stack supports lax versioning: that is, it does not throw exceptions for new unknown data members in received data.

## SOAP OPERATIONS

The WS control offers a number of methods to initiate transactions, connect to the Payment Express Server and alter settings and retrieve information. This section details the available SOAP operations and their uses.

SubmitTransaction  
SubmitTransaction2  
UpdateCard  
GetStatus (Exception Handling)  
GetStatus2 (Exception Handling)

### SubmitTransaction / SubmitTransaction2

Add the appropriate inputs (Amount, TxnType etc) with the SubmitTransaction SOAP operation to perform a purchase, refund, authorisation or completion transaction.

The SubmitTransaction SOAP method calls/outputs can be found at <https://sec.paymentexpress.com/WS/PXWS.asmx?op=SubmitTransaction>

#### Input Elements

Element	Required	Description
Amount	Yes	Amount of transaction in 1.23 format.
dateExpiry	Yes	Expiry date of card in 4 digit MMY format. Note: do not include "/" or other delimiters.
dpsTxnRef	Yes <sup>1</sup>	Only required for refund or completion transactions.
postPassword	Yes	Combined with Username, selects Account.
postUsername	Yes	Combined with Password, selects Account.
TxnRef	Yes	Set by client to uniquely identify transaction. You will need to set this value if wanting to use GetStatus.
TxnType	Yes	Purchase, Refund, Auth, Complete, Validate.
dateStart	No	The Issue date of the customer's credit card, if Issuer requires this field to be present.
enableAvsData	No	Address Verification System property. Values are 1 (Enable Verification), 0 (Disable Verification).
issueNumber	No	The Issue Number of your credit card if Issuer requires this field to be present.
avsAction	No	Address Verification System property. Valid values are 0,1,2 & 3.
avsPostCode	No	Address Verification System property. Post Code that is listed on the customer's bank statement.

<sup>1</sup> *dpsTxnRef* is required for a Refund transaction or Complete transaction only. Must contain the DpsTxnRef returned by the original Purchase or Auth transaction to be refunded or completed.

avsStreetAddress	No	Address Verification System property. Address that is listed on the customer's bank statement.
billingId	No	Specified for token billing transactions.
cardHolderName	No	Card Holder Name as on Card.
cardNumber	No	Credit Card Number. Left justified, no embedded spaces or other delimiters.
cardNumber2	No	cardNumber2 generated by DPS when adding a card for recurring billing. Needed for rebilling transactions.
clientInfo	No	Specify client information here, eg: IP address.
clientType	No	Allows the Client type to be passed through at transaction time.
cvc2	No	Card Verification number. This number is found on the back of a credit card in the signature panel - it is different from the embossed card number and provides an additional safety check.
deviceId	No	Specify device Id if applicable.
dpsBillingId	No	The Billing Id generated by DPS when adding a card for recurring billing. Needed for rebilling transactions when you do not use your own BillingId.
enable3DSecure	No	Indicates if 3D secure is to be used for the transaction.
enableAddBillCard	No	Needed for recurring billing transactions when adding a card to the DPS system. Set element to 1 for true and 0 for false.
enablePaxInfo	No	Used for Airline Reservation Systems. Enable collection of extended booking data to go through to the acquirer if they support it.
iccData	No	Additional credit card information.
inputCurrency	No	Specify the currency here.
merchantReference	No	64 character free text field.
paRes	No	paRes (Payer authentication response) POSTed back to the TermUrl.
paxCarrier	No	Used for Airline Reservation Systems. 2 character airline identifier.
paxCarrier2	No	Used for Airline Reservation Systems. 2 character airline identifier.
paxCarrier3	No	Used for Airline Reservation Systems. 2 character airline identifier.
paxCarrier4	No	Used for Airline Reservation Systems. 2 character airline identifier.
paxClass1	No	Used for Airline Reservation Systems. Class flight information.
paxClass2	No	Used for Airline Reservation Systems. Class flight information.
paxClass3	No	Used for Airline Reservation Systems. Class flight information.
paxClass4	No	Used for Airline Reservation Systems. Class flight information.
paxDate2	No	Used for Airline Reservation Systems. Flight date information.
paxDate3	No	Used for Airline Reservation Systems. Flight date information.

paxDate4	No	Used for Airline Reservation Systems. Flight date information.
paxDateDepart	No	Used for Airline Reservation Systems. Date departing in DD/MM/YY format.
paxFareBasis1	No	Used for Airline Reservation Systems. Fare Basis flight information.
paxFareBasis2	No	Used for Airline Reservation Systems. Fare Basis flight information.
paxFareBasis3	No	Used for Airline Reservation Systems. Fare Basis flight information.
paxFareBasis4	No	Used for Airline Reservation Systems. Fare Basis flight information.
paxFlightNumber1	No	Used for Airline Reservation Systems. Flight number information.
paxFlightNumber2	No	Used for Airline Reservation Systems. Flight number information.
paxFlightNumber3	No	Used for Airline Reservation Systems. Flight number information.
paxFlightNumber4	No	Used for Airline Reservation Systems. Flight number information.
paxLeg1	No	Used for Airline Reservation Systems. Leg 1 flight information.
paxLeg2	No	Used for Airline Reservation Systems. Leg 2 flight information.
paxLeg3	No	Used for Airline Reservation Systems. Leg 3 flight information.
paxLeg4	No	Used for Airline Reservation Systems. Leg 4 flight information.
paxName	No	Used for Airline Reservation Systems. Passenger Name.
paxOrigin	No	Used for Airline Reservation Systems. Passenger Origin.
paxStopOverCode1	No	Used for Airline Reservation Systems. Stop over code flight information.
paxStopOverCode2	No	Used for Airline Reservation Systems. Stop over code flight information.
paxStopOverCode3	No	Used for Airline Reservation Systems. Stop over code flight information.
paxStopOverCode4	No	Used for Airline Reservation Systems. Stop over code flight information.
paxTicketNumber	No	Used for Airline Reservation Systems. Passenger Ticket Number. Format: AAATTTTTTTTC
paxTime1	No	Used for Airline Reservation Systems. Leg1 depart time flight information.
paxTime2	No	Used for Airline Reservation Systems. Leg2 depart time flight information.
paxTime3	No	Used for Airline Reservation Systems. Leg3 depart time flight information.
paxTime4	No	Used for Airline Reservation Systems. Leg4 depart time flight information.
paxTravelAgentInfo	No	Used for Airline Reservation Systems. Travel Agent description field.
track2	No	Extracted from Track2 of credit card.
txnData1	No	Optional Free Text.
txnData2	No	Optional Free Text.
txnData3	No	Optional Free Text.

### Output Elements

Element	Description
acquirerReco	Response code sent through by the acquiring bank.
acquirerResponseText	Response code help text sent through by the acquiring bank.
amount	Amount charged in 1.23 format.
authCode	Authorisation Code (up to 64 character alphanumeric).
authorized	Indicates if the transaction was authorized or not. Either False (0) or True (1).
billingId	Billing Id specified for a token billing transaction.
cardHolderHelpText	Any tips or hints for the CardHolder. Usually just the Response Text, associated with the ReCo
cardHolderName	Card Holder Name as on Card.
cardHolderResponseDescription	A description of the transaction error to help the CardHolder, associated with the ReCo.
cardHolderResponseText	Response Text of the transaction to help the CardHolder, associated with the ReCo.
cardName	Card used (Visa,MasterCard,Bankcard etc).
cardNumber	Masked card number (first 6 and last 2 digits).
cardNumber2	Contains the cardNumber2 (token) generated by DPS when adding a card for recurring billing.
currencyId	International numeric currency code.
currencyName	ISO currency code.
currencyRate	Depends on currency submitted and link used.
dateExpiry	Card expiry date.
dateSettlement	Date transaction will be settled to Bank Account in YYYYMMDD format. This is supported for most, but not all banks and card acquirers. If the DateSettlement is not available from the banking network, the DateSettlement will contain the current calendar date.
dpsBillingId	Contains the BillingId generated by DPS when adding a card for recurring billing.
dpsTxnRef	Unique transaction identifier returned for every transaction. Required input for Refund transactions or Complete transactions.
helpText	Transaction result help text.
iccData	Additional credit card information.
issuerCountryId	The ISO 3166-1 code for the country the credit card was issued in.

	-1 - Not checked.
	0 - Checked but country not recognized.
	N - Where n is the ISO 3166-1 standard country code (e.g. NZ=554, AU=036).
merchantHelpText	Transaction result help text.
merchantReference	Transaction reference (if submitted).
merchantResponseDescription	Transaction result help text.
merchantResponseText	Transaction result help text.
reCo	2 character response code.
responseText	Response Text associated with ReCo.
retry	If true; retry transaction, if false do not retry.
riskRuleMatch	
matchedAmount	The inclusive sum of the transaction amounts causing the rule criteria to be met.
matchedTransaction	The sum of transaction counts causing the rule criteria to be met.
name	The name associated with the rule applied.
type	The type of risk management rule applied.
action	Action taken by DPS upon the matching of the rule.
period	The period in minutes that the rule covers.
includedTransactions	Transactions outcomes included when evaluating test criteria.
amountLimit	Rule criteria amount if rule type is based on amount.
transactionCountLimit	Rule criteria count if rule type is based on transaction count.
statusRequired	If true then the result of the transaction could not be determined and you will have to call GetStatus to get the result.
txnMac	Unique card identifier.
txnRef	Set by client to uniquely identify transaction.
txnType	Transaction type (Purchase, Refund, Auth, Complete, Validate).

## UpdateCard

UpdateCard call is used to update the expiry date of a BillingId, DpsBillingId or CardNumber2 associated with a credit card. This is useful when a customer's credit card has expired and they receive a new one from their bank with the same credit card number.

### Input Elements

Element	Required	Description
postPassword	Yes	Indicates the DPS account under which the token is stored.
postUsername	Yes	Authenticates the client application.
cardDetails	Yes	An escaped XML document containing dateExpiry and billingId or dpsBillingId.

### Output Elements

Element	Description
UpdateCardResult	An escaped XML document indicating the outcome of the operation.

The input SOAP message contains an element named cardDetails. The cardDetails string value should be an XML document which has been escaped before submission. dateExpiry and either dpsBillingId, billingId or CardNumber2 must be supplied.

Request example:

```
<cardDetails>  
  <dpsBillingId>0000060000991568</dpsBillingId>  
  <dateExpiry>1111</dateExpiry>  
</cardDetails>
```

The output SOAP message contains a single element named updateCardResult. The updateCardResult string value contains escaped XML which must be unescaped in order to parse the values. The updateSucceeded value indicates the success of the update and the reco and responseText values provide further details.

Output example:

```
<updateCardResult>  
  <updateSucceeded>1</updateSucceeded>  
  <reco>OK</reco>  
  <responseText>OK</responseText>  
</updateCardResult>
```

## EXCEPTION HANDLING

The Payment Express® architecture does not provide for customer applications to "reverse" or "back out" transactions once started. Exception conditions arise when the link between the customer application and the Payment Express Host is interrupted. A response to a transaction request is not received or the StatusRequired parameter in the response is true. In this circumstance, the result of the transaction is unknown.

The transaction must not be assumed to have failed. In this case, the application must enter a "recovery" mode until the result of the transaction can be ascertained. The mechanism to perform this error recovery is easily accomplished using the procedure outlined in the GetStatus method.

### GetStatus / GetStatus2

GetStatus is used when the SubmitTransaction operation fails and the response was not received within a timeout period or the link to Payment Express Server or beyond failed while awaiting a response. In this circumstance, the result of the transaction is indeterminate. The GetStatus operation should be used to retrieve the actual result. GetStatus uses the original TxnRef values which uniquely identify the transaction to lookup the transaction at the Payment Express Host and return the results.

Example:

1. Client Application loads the Web Service Input Elements
2. Client Application uses the SubmitTransaction operation call and connects to the Web Service
3. Web Service sends transaction
4. Link between Web Service client application is interrupted
5. Client application returns indeterminate result after timeout period
6. Client Application calls the GetStatus operation
7. Client Application receives the result of the transaction

Once the client application has confirmed the result of the transaction with the Payment Express Server with the GetStatus call, normal processing of subsequent transactions using SubmitTransaction method can resume.

Ensure that this txnRef value is loaded in txnRef property before calling GetStatus. Payment Express Host maintains transaction results for at least 48 hours therefore calls to GetStatus can rely on results being available for at least this period following transmission of the original transaction.

If a GetStatus method is made for a transaction that was successfully processed by Payment Express Host and was accepted by the bank, then the GetStatus method will result in the Web Service properties being set exactly as they would have been for the original transaction result, regardless of whether the original transaction result was actually received by the client application.

The GetStatus SOAP method calls/outputs can be found at <https://sec.paymentexpress.com/WS/PXWS.asmx?op=GetStatus>

**Input Elements**

Element	Required	Description
postPassword	Yes	Combined with Username, selects Account.
postUsername	Yes	Combined with Password, selects Account.
txnRef	Yes	Value assigned to the txnRef property of the transaction to look up.

**Output Elements**

Element	Description
acquirerReco	Response code sent through by the acquiring bank.
acquirerResponseText	Response code help text sent through by the acquiring bank.
amount	Amount charged in 1.23 format.
authCode	Authorisation Code (up to 64 character alphanumeric).
authorized	Indicates if the transaction was authorized or not. Either False (0) or True (1).
billingId	Billing Id specified for a token billing transaction.
cardHolderHelpText	Any tips or hints for the CardHolder. Usually just the Response Text, associated with the ReCo
cardHolderName	Card Holder Name as on Card.
cardHolderResponseDescription	A description of the transaction error to help the CardHolder, associated with the ReCo.
cardHolderResponseText	Response Text of the transaction to help the CardHolder, associated with the ReCo.
cardName	Card used (Visa,MasterCard,Bankcard etc).
cardNumber	Masked card number (first 6 and last 2 digits).
cardNumber2	Contains the cardNumber2 (token) generated by DPS when adding a card for recurring billing.
currencyId	International numeric currency code.
currencyName	ISO currency code.
currencyRate	Depends on currency submitted and link used.
dateExpiry	Card expiry date.
dateSettlement	Date transaction will be settled to Bank Account in YYYYMMDD format. This is supported for most, but not all banks and card acquirers. If the DateSettlement is not available from the banking network, the DateSettlement will contain the current calendar date.
dpsBillingId	Contains the BillingId generated by DPS when adding a card for recurring billing.

dpsTxnRef	Unique transaction identifier returned for every transaction. Required input for Refund transactions or Complete transactions.
helpText	Transaction result help text.
iccData	Additional credit card information.
issuerCountryId	The ISO 3166-1 code for the country the credit card was issued in.
	-1 - Not checked.
	0 - Checked but country not recognized.
	N - Where n is the ISO 3166-1 standard country code (e.g. NZ=554, AU=036).
merchantHelpText	Transaction result help text.
merchantReference	Transaction reference (if submitted).
merchantResponseDescription	Transaction result help text.
merchantResponseText	Transaction result help text.
reCo	2 character response code.
responseText	Response Text associated with ReCo.
retry	If true; retry transaction, if false do not retry.
riskRuleMatch	
matchedAmount	The inclusive sum of the transaction amounts causing the rule criteria to be met.
matchedTransaction	The sum of transaction counts causing the rule criteria to be met.
name	The name associated with the rule applied.
type	The type of risk management rule applied.
action	Action taken by DPS upon the matching of the rule.
period	The period in minutes that the rule covers.
includedTransactions	Transactions outcomes included when evaluating test criteria.
amountLimit	Rule criteria amount if rule type is based on amount.
transactionCountLimit	Rule criteria count if rule type is based on transaction count.
statusRequired	If true then the result of the transaction could not be determined and you will have to call GetStatus to get the result.
txnMac	Unique card identifier.
txnRef	Set by client to uniquely identify transaction.
txnType	Transaction type (Purchase, Refund, Auth, Complete, Validate).

## REFUNDS

The Web Service is capable of handling Refund transactions; however you will need to match the original Purchase or Complete transaction for this to happen. The matching is done using the DpsTxnRef value from the response of a Purchase or Complete transaction. You are able to do multiple refund transactions to the maximum amount of the original matched transaction.

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

```
TxnType = "Refund"  
DpsTxnRef  
MerchantReference (Optional)  
Amount
```

## AUTH-COMPLETE

### Overview

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

### Authorization

Use the SubmitTransaction operation with TxnType set to "Auth" for the amount to be authorised. The Auth response contains a DpsTxnRef. The funds are not transferred from the cardholder account.

### Complete

After a successful Auth transaction, but within 7 days maximum, a "Complete" transaction must be sent containing the DpsTxnRef returned by the Auth transaction.

## TOKEN BILLING

### Overview

Token Billing allows for regular/recurring billing of a cardholder card, under the control of the merchant, without requiring the merchant to either store sensitive card data securely or to obtain credit card details every time a new payment is requested. This functionality is implemented by providing the ability for a merchant to request Payment Express to capture and store credit card number and expiry date and to link these stored details to a BillingId, DpsBillingId or CardNumber2

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 **DpsBillingId** is the same as the BillingId, but is generated by DPS 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.

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

CardNumber2 tokens are generated for all transactions once enabled by DPS (please contact your DPS account manager to discuss). The token number will be returned in the cardNumber2 property of the result.

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

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

### Setup Phase

A setup phase involves loading a card into Payment Express. Optionally the setup phase can include an online \$1.00 authorisation (Validate) transaction which will determine that the card is valid and not on hot or stolen card lists and that it has the correct expiry date.

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

To add a card for future rebilling, call the SubmitTransaction web service method/operation with the following properties -

*TxnType* property set to "Purchase" or "Auth"

*EnableAddBillCard* property set to "1" (true)

*CardHolderName* (optional, strongly recommended)

*MerchantReference*

*CardNumber*

*DateExpiry*

*BillingId* (optional). If none is supplied then a DpsBillingId determined by Payment Express will be returned for use)

### Rebill Phase

The merchant application or Batch processor requests a new transaction and supplies the appropriate BillingId, DpsBillingId or CardNumber2, a MerchantReference which appears on reports and the amount to be charged. EnableAddBillCard value will be set to 0 (False) for the rebill phase.

Payment Express retrieves the credit card number and expiry date stored in the Setup Phase and a purchase transaction is formatted and processed to the card acquirer.

### Recurring Transactions

If transactions are being processed to an acquirer that supports and is configured for recurring transactions, the DPS account can also be setup to process recurring transactions only.

The main advantage of recurring transactions is that the ExpiryDate is not required. This further reduces the amount of data that needs to be stored for a merchant, and bypasses the issue of expired cards.

To setup your DPS account for recurring transaction processing only, please contact [support@paymentexpress.com](mailto:support@paymentexpress.com) (note: please ensure that your merchant bank account has been setup for recurring transactions).

## EXTENDED AIRLINE BOOKING DATA

The Web Service is capable of taking extended booking information, which is used to display on cardholders statements.

If you would like to add booking information to your transaction details you will need to set the EnablePaxInfo input property to true (1) and you will be able to use the following properties -

PaxDateDepart, PaxName, PaxLeg1, PaxLeg2, PaxLeg3, PaxLeg4, PaxOrigin, PaxTicketNumber, PaxCarrier and PaxTravelAgentInfo.

Sample data:

PaxDateDepart = "14122008"

PaxName = "Brian Smith"

PaxOrigin = "AKL"

PaxLeg1 = "SYD"

PaxLeg2 = "LAX"

PaxLeg3 = "LHR"

PaxLeg4 = "AKL"

PaxTicketNumber = "0030458343"

PaxCarrier = "QB"

PaxTravelAgentInfo = "BA1234567890"

## TEST CARDS

The following pre-approved 'test card' numbers can be used for testing, within test environments.

Visa - 4111111111111111

MasterCard - 5431111111111111

Amex - 3711111111111114

Diners - 36000000000008

Note: These can be used with any current expiry, and are suitable only for DPS test accounts.

## PROPERTIES DESCRIPTION

The following section provides a detailed description of each Web Service element and indicates if it used as an input or output. If a property is marked as input, it is not updated or output when a call to SubmitTransaction or GetStatus returns. This is important when a call is made to GetStatus for example, the original contents of Amount as input to the SubmitTransaction call are not output by GetStatus.

**AcquirerReco (output)** Datatype: String Max 2 characters  
2 character response code from the Acquiring bank.

**AcquirerResponseText (output)** Datatype: String Max 32 characters  
Response Text associated with ReCo from the Acquiring bank.

**Amount (input/output)** Datatype: String Max 12 characters

Total Purchase, Refund, Auth or Complete 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. Maximum value allowable is \$99,999.99. Note that acquirer or card limits may be lower than this amount. When submitting transactions for currencies with no decimal division of units such as JPY the AmountInput must be in an appropriate format e.g. "10".

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

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

**AvsAction (input)** 1 digit  
Address Verification System property. Valid values are 0, 1, 2 & 3.

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

The value will most likely be 1 for most circumstances.

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

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

**BillingId (input/output)** Datatype: String Max 32 characters  
BillingId generated by the customer system. This could be a customer number and is used as input to SubmitTransaction to rebill an existing customer. If EnableAddBillCard

**CardHolderHelpText (output)** Data type: String Max 32 characters  
More detailed explanation of result. Intended for card holder.

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

**CardHolderResponseDescription (output)** Data type: String Max 32 characters  
More detailed explanation of result. Intended for card holder.

**CardHolderResponseText (output)** Data type: String Max 32 characters  
Brief (Max 32 character) response text intended for card holder.

**CardName (output)** Datatype: String Max 16 characters  
The card type used for the transaction. Note that the list may be expanded as support for new cards is added. The CardName format is to capitalize the first letter with remaining letters in lowercase.

CardName Value	Description
Amex	American Express
Bankcard	Bank Card
Diners	Diners Card
Jcb	JCB
Mastercard	Mastercard
Visa	Visa

**CardNumber (input/output)** Datatype: String Max 19 characters

The card number. No leading or embedded blanks are permitted. Must contain a numeric value. Not required for Complete or Refund TxnType.

**CardNumber2 (input/output)** Datatype: String Max 19 characters

A Billing Token value formatted to resemble a card number and pass a Luhn check. To use CardNumber2 tokens your account must be configured to generate them. Please contact our support team if you intend to use this feature.

**ClientInfo (input)** Datatype: String Max 64 characters

Optional field to send through client information, usually an IP address.

**ClientType (input)** Datatype: String

Allows the Client type to be passed through at transaction time. Default value if not submitted is set to "I".

Allowed values:

Value	Description
V	IVR
m	MOTO
U	Unattended
I	Internet
Recurring	Recurring*

\* Recurring, if supported by your merchant bank expiry date will not be validated.

**CurrencyId (output)** Datatype: INT Max 4 characters

Based on input/default currency.

**CurrencyName (output)** Datatype: String Max 4 characters

Currency code, based on input/default currency.

**CurrencyRate (output)** Datatype: String Max 4 characters

Based on input/default currency.

**Cvc2 (input)** Datatype: String Max 4 characters

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

American Express: Four-digit batch code (4DBC)

MasterCard: Card Verification Code 2 (CVC2)

Visa: Card Verification Value 2 (CVV2)

Supplying this value provides an indication of that the person participating in a transaction had physical possession of the card at some point in time. This is not currently implemented by all acquirer and may not necessarily be checked.

**DateExpiry (input/output)** Datatype: String Max 4 characters

Indicates card expiry date. Format is MMYT where MM is month 01-12 and Year 00-99. do not insert "/" or other delimiter. Not required for Complete or Refund transactions.

**DateSettlement (output)** Datatype: String Max 8 characters

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

**DateStart (input)** Datatype: String Max characters

The Issue date of the customer's credit card, if Issuer requires this field to be present.

Format is MMYT where MM is month 01-12 and Year 00-99. do not insert "/" or other delimiter.

Used for Maestro/Solo cards.

**Deviceld (input)** Datatype: String Max 32 characters

Optional field where a device specific ID can be specified.

**DpsBillingId (input/output)** Datatype: String Max 16 characters

Returned for a successful billing transaction if EnableAddBillCard is set. Supplied as input to rebill a transaction if BillingId is not used. It is not allowed to specify both a BillingId and a DpsBillingId when rebilling a transaction.

**DpsTxnRef (input/output)** Datatype: String Max 16 characters

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

**Enable3Dsecure (input)** Data type: Boolean True/False (1 or 0)

Indicates if 3D secure is to be used for the transaction. Value will need to be true (1) if 3D secure information is included with the transaction.

**EnableAddBillCard (input)** Datatype: Boolean True/False (1 or 0)

If set to true (1) on input to SubmitTransaction, the details necessary to charge the same customer in the future are securely stored. A BillingId may optionally be attached on input. DpsBillingId is returned. See Token Billing section for details on using this feature.

**EnableAvsData (input)** Datatype: INT

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

**EnablePaxInfo (input)** Data type: Boolean True/False (1 or 0)

Used for Airline Reservation Systems. Enable collection of extended booking data to go through to the acquirer. Value will need to be true (1) if ticket information is included with the transaction.

**HelpText (output) Data type:** String Max 255 characters

Transaction result help text.

**lccData (input/output)** Data type: String Max 16 characters

Additional credit card information.

**InputCurrency (input)** Datatype: String Max 4 characters

Indicates currency used for this transaction. If blank, currency will be determined by the bank account used which is selected using the Username/Password details. Not all acquirers can support multiple currencies. Valid values for Currency are:

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

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

**IssueNumber (input)** Datatype: INT

The Issue Number of your credit card if Issuer requires this field to be present.

**IssuerCountryId (output)** Datatype: INT

The Country code for the country the credit card used was issued in.

-1 - Not checked

0 - Checked but country not recognized possibly invalid bin range or invalid card

N - Where n is the ISO 3166-1 standard country code (e.g. NZ=554, AU=036)

**MerchantHelpText (output)** Datatype: String Max 16 characters

Transaction result help text.

**MerchantReference (input/output)** Datatype: String Max 64 characters

Free Text Field for use by merchant (could be order number, customer number etc.).

**MerchantResponseDescription (output)**

Transaction result help text.

**MerchantResponseText (output)** Datatype: String Max 16 characters

Transaction result help text.

**PaRes (input)** Data type: String Max 16 characters

Used for 3D secure transactions. Payer authentication response. Value to be included in the SubmitTransaction message.

**PaxCarrier (input)** Data type: String Max 2 characters

Used for Airline Reservation Systems. 2 character airline identifier. Alphanumeric.

**PaxCarrier2 (input)** Data type: String Max 2 characters

Used for Airline Reservation Systems. 2 character airline identifier. Alphanumeric.

**PaxCarrier3 (input)** Data type: String Max 2 characters

Used for Airline Reservation Systems. 2 character airline identifier. Alphanumeric.

**PaxCarrier4 (input)** Data type: String Max 2 characters

Used for Airline Reservation Systems. 2 character airline identifier. Alphanumeric.

**PaxClass1 (input)** Data type: String Max 1 characters

Used for Airline Reservation Systems. Class flight information. Alphanumeric.

**PaxClass2 (input)** Data type: String Max 1 characters

Used for Airline Reservation Systems. Class flight information. Alphanumeric.

**PaxClass3 (input)** Data type: String Max 1 characters

Used for Airline Reservation Systems. Class flight information. Alphanumeric.

**PaxClass4 (input)** Data type: String Max 1 characters

Used for Airline Reservation Systems. Class flight information. Alphanumeric.

**PaxDateDepart (input)** Data type: String Max 8 characters

Used for Airline Reservation Systems. Date departing in YYYYMMDD format. Numeric.

**PaxDate2 (input)** Data type: String Max 8 characters

Used for Airline Reservation Systems. Date departing in YYYYMMDD format. Numeric.

**PaxDate3 (input)** Data type: String Max 8 characters

Used for Airline Reservation Systems. Date departing in YYYYMMDD format. Numeric.

**PaxDate4 (input)** Data type: String Max 8 characters  
Used for Airline Reservation Systems. Date departing in YYYYMMDD format. Numeric.

**PaxFlightNumber1 (input)** Data type: String Max 6 characters  
Used for Airline Reservation Systems. Flight number 1 information. Alphanumeric.

**PaxFlightNumber2 (input)** Data type: String Max 6 characters  
Used for Airline Reservation Systems. Flight number 2 information. Alphanumeric.

**PaxFlightNumber3 (input)** Data type: String Max 6 characters  
Used for Airline Reservation Systems. Flight number 3 information. Alphanumeric.

**PaxFlightNumber4 (input)** Data type: String Max 6 characters  
Used for Airline Reservation Systems. Flight number 4 information. Alphanumeric.

**PaxLeg1 (input)** Data type: String Max 3 characters  
Used for Airline Reservation Systems. Leg 1 flight information. Alphanumeric.

**PaxLeg2 (input)** Data type: String Max 3 characters  
Used for Airline Reservation Systems. Leg 2 flight information. Alphanumeric.

**PaxLeg3 (input)** Data type: String Max 3 characters  
Used for Airline Reservation Systems. Leg 3 flight information. Alphanumeric.

**PaxLeg4 (input)** Data type: String Max 3 characters  
Used for Airline Reservation Systems. Leg 4 flight information. Alphanumeric.

**PaxName (input)** Data type: String Max 20 characters  
Used for Airline Reservation Systems. Passenger Name. Alphanumeric.

**PaxOrigin (input)** Data type: String Max 3 characters  
Used for Airline Reservation Systems. Passenger Origin of departure. Alphanumeric.

**PaxStopOverCode1 (input)** Data type: String Max 1 characters  
Used for Airline Reservation Systems. Stop over code flight information. Alphanumeric.

**PaxStopOverCode2 (input)** Data type: String Max 1 characters  
Used for Airline Reservation Systems. Stop over code flight information. Alphanumeric.

**PaxStopOverCode3 (input)** Data type: String Max 1 characters  
Used for Airline Reservation Systems. Stop over code flight information. Alphanumeric.

**PaxStopOverCode4 (input)** Data type: String Max 1 characters  
Used for Airline Reservation Systems. Stop over code flight information. Alphanumeric.

**PaxTicketNumber (input)** Data type: String Max 10 characters  
Used for Airline Reservation Systems. Passenger Ticket Number. Format: AAATTTTTTTTTTC. AAA is airline code, TTTTTTTTTT (10 chars) is actual ticket number and C is check digit. Numeric.

**PaxTime1 (input)** Data type: String Max 4 characters  
Used for Airline Reservation Systems. Leg1 depart time flight information. Alphanumeric.

**PaxTime2 (input)** Data type: String Max 4 characters  
Used for Airline Reservation Systems. Leg2 depart time flight information. Alphanumeric.

**PaxTime3 (input)** Data type: String Max 4 characters  
Used for Airline Reservation Systems. Leg3 depart time flight information. Alphanumeric.

**PaxTime4 (input)** Data type: String Max 4 characters  
Used for Airline Reservation Systems. Leg4 depart time flight information. Alphanumeric.

**PaxTravelAgentInfo (input)** Data type: String Max 25 characters  
Used for Airline Reservation Systems. Travel Agent description field. Also known as the Booking Reference on some of DPS screens. Alphanumeric free text field.

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

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

**ReCo (output)** Datatype: String Max 2 characters

Response Code generated by DPS Server (for locally declined transactions) or by the Card Acquirer (for host originated responses). The ReCo should not be checked by the client application, as these values differ according to acquirer. Use the Success property to check for successful finishing of a transaction. Refer to the Response Codes section for a list of valid response code errors generated by Payment Express.

**ResponseText (output)** Datatype: String Max 32 characters

Response Text associated with the response code of the transaction

**Retry (output)** Datatype: Boolean True/False (1 or 0)

If true, then the transaction should be sent again - the transaction was declined due to a temporary error. If false, then the transaction should not be sent again, regardless of whether it was accepted or declined.

**RiskRuleMatches (output):**

**MatchedAmount** - The inclusive sum of the transaction amounts causing the rule criteria to be met.

**MatchedTransactionCount** - The sum of transaction counts causing the rule criteria to be met.

**Name** - The name associated with the rule applied.

**Type** - The type of risk management rule applied.

**Action** - Action taken by DPS upon the matching of the rule.

**Period** - The period in minutes that the rule covers.

**IncludedTransactions** - Transactions outcomes included when evaluating test criteria.

**AmountLimit** - Rule criteria amount if rule type is based on amount.

**TransactionCountLimit** - Rule criteria count if rule type is based on transaction count.

**StatusRequired (output)** Datatype: Boolean True/False (1 or 0)

If true then the status of the transaction could not be determined, either because the server could not be reached, because of an error, or because the transaction is still in progress. Your application should call GetStatus until StatusRequired is false (0).

**Track2 (input)** Datatype: String

Extracted from Track2 of the credit card.

**TxnData1, TxnData2, TxnData3 (input):** String Max 255 characters

Optional free text fields. Usually assigned at origin website.

**TxnMac (output)** Datatype: String Max 16 characters

Unique value generated based on credit card number.

**TxnRef (input / output)** Datatype: String Max 16 characters

An identifier provided by your application to uniquely identify the transaction. This is the TxnRef supplied by the client to initiate the transaction, or, if not supplied, a TxnRef value internally generated by DPS on return. This is required as an input for GetStatus operations.

**TxnType (input/output)** Datatype: String Max 8 character

Value	Meaning
Auth	Authorise - amount is authorised, no funds transferred.
Complete	Complete a previous authorization - funds are transferred.
Purchase	Purchase - Funds are transferred immediately.
Refund	Refund - Funds transferred immediately. Must be enabled as a special option.
Validate	Validation Transaction. Sends through a \$1.00 Auth to validate card details including expiry date. Often utilised with the EnableAddBillCard property set to 1 to automatically add to Billing Database if the transaction is approved (validate). Note that the Validate transaction type is not enabled by default on live accounts. Please make a request to DPS Support if you would like to utilise this transaction type