DIRECT

INTEGRATION GUIDE

Version: 10.00

1 Direct HTTP Integration

1.1 About This Guide

The Direct HTTP integration works by allowing you to keep the Customer on your system through the checkout process while processing the transactions via the Gateway in the background. This allows you to provide a smoother, more complete checkout process to the customer.

If you wish to take card details on your website, or style your payment pages, then you either need to use the Direct integration or use the Hosted integration and request a Custom Hosted Payment Page for your website.

To use the Direct integration your website must have a SSL Certificate. You will also need to consider the Payment Card Industry Data Security Standard (PCI:DSS) when capturing card details. For more information, please see https://www.pcisecuritystandards.org/.

In addition to transaction processing, the Direct integration can be used to perform other actions such as refunds and cancellations which can provide a more advanced integration with the Gateway.

This guide provides the information required to integrate with the Payment Gateway and gives a very basic example of code for doing so (further examples can be found on our website). It is expected that you have some experience in server side scripting with languages such as PHP or ASP, or that an off-the-shelf software package is being used that has in-built or plug-in support for the Payment Gateway.

If you do require programming assistance related to your integration, please contact FideliPay on 0845 223 4935 or via email to support@fidelipay.co.uk.

1.2 Integration Disclaimer

FideliPay provides all integration documentation necessary for enabling Merchants to process payments via our Payment Gateway. Whilst every effort has been made to ensure these guides are accurate and complete, we expect Merchants undertaking any integration to test all their technical work fully and satisfy their own standards. FideliPay is not responsible or liable for any Merchant or Third Party integration.

1.3 Terminology

The following terms are used throughout this guide;

Gateway
The FideliPay Payment Gateway

Merchant
The Merchant using the Gateway's services

Acquirer
The bank or financial institution used by the Merchant.

Customer
A customer of the Merchant making a payment etc.

Cardholder
The person who owns the payment card, normally the Customer.

Merchant Account
An account on the Gateway mapped to an Acquirer issued account.

You/your
The Merchant or their representative performing the integration.

1.4 Pre-Requisites

You will need the following information to integrate with the Payment Gateway using the Direct integration method;

FideliPay Merchant ID	Your Merchant ID enables you to access and communicate with the Payment Gateway. Please note that these details will differ to the login supplied to access the administration panel. You should have received these details when your account was set up. You may also use test Merchant IDs (if you have been issued with a test ID) and swap these for your live account details when you receive them.
Integration URL	https://gateway.fidelipay.co.uk/direct/

New Merchants who have not yet received their live Merchant ID can still perform an integration for testing purposes. Simply enter one of the test Merchant IDs below and use the test cards provided in appendix A-10 to run a test transaction.

For non 3-D Secure testing use Merchant ID **101093**
For 3-D Secure Testing use Merchant ID **101094**

1.5 Integration Details

1.5.1 Direct Requests

A request can be sent to the Gateway by submitting a HTTP POST request to the integration URL provided.

The request should be URL encoded as `name=value` fields separated by '&' characters. The response will be received in the same format.

Example URL encoding:

```
merchantID=101093&action=SALE&type=1&amount=1001&currencyC
ode=826&countryCode=826&transactionUnique=55f6db1c81d95&o
rderRef=Test+purchase&customerPostCode=NN17+8YG&responseC
ode=0&responseMessage=AUTHCODE%3A350333&state=captured&xr
ef=15091702MG47WN32MM88LPK&cardNumber=4929+4212+3460+0821
&cardExpiryDate=1215
```

For more information on the URL encoded format refer to RFC 1738 and the `application/x-www-form-urlencoded` media type.

Please note that the field names are cAsE sEnSiTiVe.

The response will return the request fields in addition to any dedicated response field. If the request contains a field that is also intended as a response field then any incoming request value will be overwritten by the correct response value.

1.5.2 Callback URL

You can request that the Gateway sends a copy of the response to an alternative URL using the **`callbackURL`** request field. In this case each response will be then POSTed to that URL in addition to the normal response. This allows you to specify a URL on a secure shopping cart or backend order processing system which will then fulfil any order etc. related to the transaction.

1.6 Authentication

All requests must specify which Merchant Account they are for using the **`merchantID`** request field. In addition to this the following security measures can be used;

1.6.1 Password Authentication

You can configure a password for each Merchant Account using the Merchant Management System (MMS). This password must then be sent in the **`merchantPwd`** field in each request. I an incorrect password is received by the Gateway then the transaction will be aborted and an error response returned

> Warning: Use of a password is discouraged in any integration where the transaction is posted from a form in the client browser as the password may appear in plain text in code.

1.6.2 Message signing

Message signing requires you to generate a hash of the request message being sent and then send this hash along with the original request in the **`signature`** field. The gateway will then re-generate the hash on the request message received and compare it with the one sent. If the two hashes are different then the request received must not be the same as that sent and so the contents must have been tampered with and the transaction will be aborted and an error response returned

The gateway will also return hash of the response message in the returned **`signature`** field allowing the merchant to create a hash of the response (minus the **`signature`** field) and verify the hashes match.

If message signing is enabled, then the data POSTed to any callback URL will also be signed.

See appendix A-11 for information on how to create the hash.

1.6.3 Allowed IP addresses

You can configure a list of IP addresses using the Merchant Management System (MMS). Two different address lists can be configured, one for standard requests, such as sales, and one for advanced requests, such as refunds and cancellations. If a request is received from an address other than those configured, then it will be aborted and an error response returned.

1.7 Supported Actions

All requests must specify what action they require the Gateway to perform using the **`action`** request field. The Direct integration allows the following actions to be specified;

1.7.1 SALE

This will create a new transaction and attempt to seek authorisation for a sale from the Acquirer. A successful authorisation will reserve the funds on the cardholder's account until the transaction is settled.

The **`captureDelay`** field can be used to state if the transaction should be authorised only and settled at a later date. For more details on delayed capture refer to appendix A-8.

1.7.2 VERIFY

This will create a new transaction and attempt to verify that the card account exists with the Acquirer. The transaction will result in no transfer of funds and no hold on any funds on the cardholder's account. It cannot be captured and will not be settled. The transaction **`amount`** must always be zero.

This transaction type is the preferred method for validating that the card account exists and is in good standing, however it cannot be used to validate that it has sufficient funds.

1.7.3 PREAUTH

This will create a new transaction and attempt to seek authorisation for a sale from the Acquirer. If authorisation is approved, then it is immediately voided (where possible) so that no funds are reserved on the cardholder's account. The transaction will result in no transfer of funds. It cannot be captured and will not be settled.

This transaction type can be used to check whether funds are available and that the account is valid. However due to the problem highlighted below it is recommended that Merchants use the VERIFY when supported by their Acquirer.

Warning: If the transaction is to be completed then a new authorisation needs to be sought using the SALE action. If the PREAUTH authorisation could not be successfully voided then this will result in the funds being authorised twice effectively putting 2 holds on the amount on the cardholder's account and thus requiring twice the amount to be available in the cardholder's account. It is therefore recommended to only PREAUTH small amounts such as £1 to mainly check account validity.

1.7.4 REFUND_SALE

This will create a new transaction and attempt to seek authorisation for a refund of a previous SALE from the Acquirer. The transaction will then be captured and settled if and when appropriate. It can only be performed on transactions that have been successfully settled, up until that point a CANCEL or partial CAPTURE can be done to refund or partially refund the original SALE transaction. The previous SALE transaction should be specified using the **`xref`** field.

Partial refunds are allowed by specifying the **`amount`** to refund, any amount must not be greater than the original received amount minus any already refunded amount. Multiple partial refunds may be made while there is still a portion of the originally received amount un-refunded.

The **`captureDelay`** field can be used to state if the transaction should be authorised only and settled at a later date. For more details on delayed capture refer to appendix A-8.

1.7.5 REFUND

This will create a new transaction and attempt to seek authorisation for a refund from the Acquirer. The transaction will then be captured and settled if and when appropriate. This is an independent refund and need not be related to any previous SALE. The amount is therefore not limited by any original received amount.

The **`captureDelay`** field can be used to state if the transaction should be authorised only and settled at a later date. For more details on delayed capture refer to appendix A-8.

1.7.6 CAPTURE

This will capture an existing transaction, identified using the **`xref`** request field, making it available for settlement at next available opportunity. It can only be performed on transactions that have been authorised but not yet captured. An **`amount`** to capture may be specified but must not exceed the original amount authorised.

The original transaction must have been submitted with a **`captureDelay`** value that prevented immediate capture and settlement leaving the transaction in an authorised but un-captured state. For more details on delayed capture refer to appendix A-8.

1.7.7 CANCEL

This will cancel an existing transaction, identified using the **`xref`** request field, preventing it from being settled. It can only be performed on transactions, which have been authorised but not yet settled, and it is not reversible. Depending on the Acquirer it may not reverse the authorisation and release

any reserved funds on the cardholder's account, in such cases authorisation will be left to expire as normal releasing the reserved funds – this may take up to 30 days from the date of authorisation.

1.7.8 QUERY

This will query an existing transaction, identified using the **`xref`** request field, returning the original response. This is a simple transaction lookup action.

1 New Transactions

You can perform a new transaction, such as a sale, by sending a request with the required action and transaction type along with details about the order and payment method.

1.1 Request Fields

Field Name	Mandatory?	Description
merchantID	Yes	Your Gateway Merchant ID.
merchantPwd	No[1]	Any password used to secure this account. Refer to section 1.6.1 for details.
signature	Yes[2]	Any hash used to sign this request. Refer to section 1.6.2 for details.
action	Yes	The action requested. Refer to section 1.7 for supported actions. Possible values are: **PREAUTH, SALE, REFUND, REFUND_SALE, VERIFY.**
amount	Yes[3]	The amount of the transaction in minor currency. For the UK, this is pence, so £10.99 should be sent as 1099. **Numeric values only – no decimal points or currency symbols.**
type	Yes[3]	The type of transaction. Refer to section A-13 for details. Possible values are: **1** – E-commerce (ECOM) **2** - Mail Order/Telephone Order (MOTO). **9** – Continuous Authority (CA).
countryCode	Yes[3]	Merchant's location. **Valid ISO-3166 alpha or numeric code.**
currencyCode	Yes[3]	Transaction currency. **Valid ISO-4217 alpha or numeric code.**
cardNumber	Yes[3]	The primary account number (PAN) as printed on the front of the payment card. **Numeric values only (spaces allowed).**
cardExpiryMonth	Yes[3]	Payment card's expiry month as a number from 1 to 12. **Numeric values only.**
cardExpiryYear	Yes[3]	Last two digit of the payment card's expiry year as a number from 00 to 99.

Field Name	Mandatory?	Description
		Numeric values only.
cardCVV	Yes[3]	Payment card's security number. The 3 digit number printed on the payment cards signature strip[4]. **Numeric values only.**
cardExpiryDate	No[3]	Payment card's expiry date in MMYY format as an alternative to sending separate **cardExpiryMonth** & **cardExpiryYear** fields. **Numeric values only.**
transactionUnique	No[3]	You can supply unique identifier for this transaction. This is an added security feature to combat transaction spoofing.
orderRef	No[3]	Free format test field to store order details, reference numbers, etc. for the Merchant's records.
captureDelay	No	Number of days to wait between authorisation of a payment and subsequent settlement. Refer to appendix A-8 for details.
xref	No[5]	Reference to a previous transaction. Refer to appendix A-14 for details.
callbackURL	No	A non-public URL which will receive a copy of the transaction result by POST. Refer to section 1.5.2 for details.

If the REFUND_SALE action is used, then the request may not attempt to change the payment details or the request will fail with a **responseCode** of **65542 (REQUEST MISMATCH)** because the refund must be made to the original card.

[1] A password is not recommended if using the Hosted Integration, use a signature instead.
[2] A signature is recommended if using the Hosted Integration.
[3] Optional if an **xref** is provided as the value will be taken from the cross referenced transaction.
[4] For American Express cards this is a 4 digit number printed flat on the front of the card.
[5] Mandatory for a REFUND_SALE request to specify the original SALE transaction.

2.2 Response Fields

The response will contain all the fields sent in the request (minus any card details) plus the following;

Field Name	Returned?	Description
responseCode	Always	A numeric code providing the outcome of the transaction: Possible values are: **0** - Successful / authorised transaction. **2** - Card referred. **4** - Card declined – keep card. **5** - Card declined. Check **responseMessage** for more details of any error that occurred. Refer to appendix A-1 for details.
responseMessage	Always	The message received from the Acquiring bank, or any error message.
transactionID	Always	A unique ID assigned by the Gateway.
xref	Always	You may store the cross reference for repeat transactions. Refer to appendix A-14 for details.
state	Always	Transaction state. Refer to appendix A-12.2 for details.
timestamp	Always	Time the transaction was created or last modified.
transactionUnique	If supplied	Any value supplied in the initial request.
authorisationCode	On success	Authorisation code received from Acquirer.
referralPhone	If provided	Telephone number supplied by Acquirer to phone for voice authorisation. Most Acquirers do not provide this number.
amountReceived	On success	The amount the Acquirer authorised. This should always be the full **amount** requested.
amountRefunded	If refund	Total amount of original SALE that has so far been refunded. Returned when **action** is REFUND_SALE.
orderRef	If supplied	Any value supplied in the initial request.
cardNumberMask	Always	Card number masked so only the last 4 digits are visible.

Field Name	Returned?	Description
cardTypeCode	Always	The code of card used. Refer to appendix A-9 for details.
cardType	Always	The description of the card used. Refer to appendix A-9 for details.
cardSchemeCode	Always	The code of the card scheme used. Refer to appendix A-9 for details.
cardScheme	Always	The description of the card scheme used. Refer to appendix A-9 for details.
cardIssuer	Always	The card issuer (when known).
cardIssuerCountry	Always	Name of card issuing country (when known).
cardIssuerCountryCode	Always	ISO-3166 Alpha 2 code of the card issuing country (when known)

Note: the response is also POSTed to any URL provided by optional **callbackURL**.

1 Management Requests

You can perform an action on an existing transaction, such as a capture or cancellation, by sending a request with the required action along with the cross reference for the transaction to act on.

1.1 Request Fields

Field Name	Mandatory?	Description
merchantID	Yes	Your Gateway Merchant ID.
merchantPwd	No[1]	Any password used to secure this account. Refer to section 1.6.1 for details.
signature	Yes[2]	Any hash used to sign this request. Refer to section 1.6.2 for details.
action	Yes	The action requested. Refer to section 1.7 for supported actions. Possible values are: **REFUND_SALE, CAPTURE, CANCEL, QUERY**.
xref	Yes	Reference to a previous transaction. Refer to appendix A-14 for details.
amount	No[3]	The amount of the transaction in minor currency. For the UK, this is pence, so £10.99 should be sent as 1099. **Numeric values only – no decimal points or currency symbols.**
callbackURL	No	A non-public URL which will receive a copy of the transaction result by POST. Refer to section 1.5.2 for details.

[1] A password is not recommended if using the Hosted Integration, use a signature instead.
[2] A signature is recommended if using the Hosted Integration.
[3] An amount is only required for partial refunds or partial captures.

3.2 Response Fields

With the exception of the fields below, the response will be the same as for a new transaction, but will contain the details of the existing transaction.

Field Name	Returned?	Description
responseCode	Always	A numeric code providing the outcome of the management request. Check **responseMessage** for more details of any error that occurred. Refer to appendix A-1 for details.
responseMessage	Always	Description of above response code.
action	Always	The requested action and original action separated by a colon. For example. **CANCEL:SALE**

1 AVS/CV2 Checking

1.1 Background

You are able to request AVS and CV2 fraud checking on transactions processed by the Payment Gateway.

These fraud prevention checks are performed by the Acquirer while authorising the transaction. You can choose how to act on the outcome of the check (or even to ignore them altogether).

1.1.1 AVS Checking

The Address Verification System (AVS) uses the address details that are provided by the cardholder to verify the address is registered to the card being used. The address and postcode are checked separately.

1.1.2 CV2 Checking

CV2, CVV, or Card Verification Value is a 3 or 4 digit security code –The check verifies the code is the correct one for the card used.

For most cards the CVV is a 3 digit number to the right of the signature strip. For American Express cards this is a 4 digit number printed, not embossed, on the front right of the card.

The AVS/CV2 checking preferences can be configured per Merchant Account within the Merchant Management System (MMS). These preferences can be overridden per transaction by sending one of the preference fields documented in section 4.3 which hold a comma separated list of the check responses that should be allowed to continue to completion. Responses not in the list will result in the transaction being declined with a **`responseCode`** of **5 (AVS/CV2 DECLINED)**.

1.2 Benefits & Limitations

1.2.1 Benefits

- Instant: The results are available immediately and returned as part of the transaction
- Flexible: The checks can be managed independently allowing you the upmost control over how the results are used.
- Automatic: The checks can be configured to automatically decline transaction where required.

1.2.2 Limitations

- Not all countries supported: AVS is a UK scheme only: It is not possible to check AVS on non-UK issued cards.
- Only Address numerics are checked: The non-numerical characters in the billing address and postcode are not checked as part of the AVS checks.
- Unable to check AVS/CV2 on company cards: If you accept company credit cards you are not able to receive results on all company cards. This is due to the Acquirers not having access to this information

1.3 Request Fields

These fields should be sent in addition to basic request fields in section 2.1.

Field Name	Mandatory?	Description
customerAddress	Yes[1]	For AVS checking this must be a registered billing address for the card.
customerPostCode	Yes[2]	For AVS checking this must be a registered postcode for the card.
cardCVV	Yes[3]	For CVV checking this must be the Card Verification Value printed on the card.
avscv2CheckRequired	No[4]	Is AVS/CV2 checking required for this transaction? Possible values are: **N** – Checking is not required. **Y** – Abort if checking is not enabled.
cv2CheckPref	No[5]	List of **cv2Check** response values that are to be accepted, any other value will cause the transaction to be declined. Value is a comma separated list containing one or more of the following**: not known, not checked, matched, not matched, partially matched**.
addressCheckPref	No[5]	List of **addressCheck** values that are to be accepted, any other value will cause the transaction to be declined. Value is a comma separated list containing one or more of the following **not known, not checked, matched, not matched, partially matched**.
postcodeCheckPref	No[5]	List of **postcodeCheck** response values that are to be accepted, any other value will cause the transaction to be declined Value is a comma separated list containing one or more of the following: **not known, not checked, matched, not matched, partially matched**.

[1] Mandatory if AVS address checking is required
[2] Mandatory if AVS postcode checking is required
[3] Mandatory if CV2 checking is required
[4] The default value is **Y** if AVS/CV2 checking is enabled on the Merchant Account, otherwise **N**
[5] If the value is not supplied than the default account preferences will be used.

1.1 Response Fields

These fields will be returned in addition to the AVS/CV2 request fields in section 4.3 the basic response fields in section 0.

Field Name	Returned?	Description
avscv2CheckEnabled	Always	Is AVS/CV2 checking enabled for this Merchant Account? Possible values are: **N** – Merchant account is not enabled. **Y** – Merchant account is enabled.
avscv2ResponseCode	If checks performed	The result of the AVS/CV2 check. Refer to appendix A-2 for details.
avscv2ResponseMessage	If checks performed	The message received from the Acquiring bank, or any error message with regards to the AVS/CV2 check. Refer to appendix A-2 for details.
avscv2AuthEntity	If checks performed	Textual description of the AVS/CV2 authorizing entity as described in appendix A-2. Possible values are: **not known, merchant host, acquirer host, card scheme, issuer.**
cv2Check	If checks performed	Description of the AVS/CV2 CV2 check as described in appendix A-2. Possible values are**: not known, not checked, matched, not matched, partially matched**.
addressCheck	If checks performed	Description of the AVS/CV2 address check as described in appendix A-2. Possible values are**: not known, not checked, matched, not matched, partially matched**.
postcodeCheck	If checks performed	Description of the AVS/CV2 postcode check as described in appendix A-2. Possible values are**: not known, not checked, matched, not matched, partially matched**.

1 3-D Secure Authentication

1.1 Background

3-D Secure authentication is an additional fraud prevention scheme that is available to all Merchants using the Payment Gateway.

It allows Cardholder's to assign a password to their card that is then verified whenever a transaction is processed through a site that supports the use of the scheme. The addition of password protection allows extra security on transactions that are processed online.

3-D Secure stands for 3 Domain Server, there are 3 parties that are involved in the 3-D Secure process:

- The company the purchase is being made from.
- The Acquiring Bank (the bank of the company)
- VISA and MasterCard (the card issuers themselves)

The gateway supports 3-D Secure as implemented by Visa and Mastercard and marketed under the brand names of Verified by VISA (VBV) and MasterCard Secure Code (MSC). Implementations by American Express (SafeKey) and JCB (J/Secure) are not currently supported.

3-D Secure is also the only fraud prevention scheme that is available that offers Merchants liability cover for transactions that are verified by the checks. This provides additional protection to Merchants using the scheme as opposed to those that do not.

1.2 Benefits & Limitations

1.2.1 Benefits

- Instant: The results are available immediately and returned as part of the transaction
- Flexible: The checks can be managed independently allowing you the upmost control over how the results are used.
- Automatic: The checks can be configured to automatically decline transaction where required.
- Liability Shift: The main benefit to companies using the 3-D Secure scheme is the availability of a liability shift for a successfully authenticated transaction. This offers protection by the card issuers against charge backs as the liability is assumed. Note: You will need to check with your Acquirer for the exact terms on liability shifts.
- No extra cost: There are no extra costs to add 3-D Secure onto your gateway account. Your Acquirer may charge to add this onto your Merchant Account however you may also find that your transaction charges lower as a result of using 3-D Secure.
- Easy management: The 3-D Secure scheme is controlled within the Merchant Management System (MMS).

1.2.2 Limitations

- Chargebacks can still occur: Fully authenticated 3-D Secure transactions do not guarantee a liability shift; this is decided on the discretion of your Acquirer.
- Not all cards are supported: At the moment the gateway does not support 3-D Secure for Amex, JCB or Diner's club cards.

1.3 Implementation

If your Merchant account is setup for 3-D Secure the Gateway will require further authentication details provided by the 3-D Secure system.

1.3.1 Initial Request (Verify Enrolment)

If no 3-D Secure authentication details are provided in the initial request the Gateway will determine if the transaction is eligible for 3-D Secure by checking if the card is enrolled in the 3-D Secure scheme.

If the Gateway determines that the transaction is not eligible for 3-D Secure then it will continue and process it as normal transaction without 3-D Secure unless the **`threeDSRequired`** request field indicates that the transaction should be aborted instead.

If the Gateway determines that the transaction is eligible it will respond with a **`responseCode`** of **65802 (3DS AUTHENTICATION REQUIRED)** and included in the response will be a **`threeDSACSURL`** field containing the URL required to contact the ACS on and a **`threeDSMD`** and **`threeDSPaReq`** to send to the provided URL. The latter two values must be posted to the provided ACS URL as the fields **MD** and **PaReq** along with a **TermUrl** field provided by the yourself which must contain the URL of a page on the Merchant's server to return to when authentication has been completed.

1.3.2 Continuation Request (Check Authentication & Authorise)

On completion of the 3-D Secure authentication the ACS will post the original **MD** along with a **PaRes** value to the **TermUrl** provided. These values should then be sent to the Gateway in the **`threeDSMD`** and **`threeDSPaRes`** fields of a new request. This new request will check the 3-D Secure authentication and then either complete or abort the transaction depending on the authentication result and the your preferences, either sent in the **`threeDSPref`** field on set in the Merchant Management System (MMS).

If you would like an example of a 3-D Secure integration, please refer to our sample code Appendix **A-17.1.**

1.3.3 External Authentication Request

You can choose to obtain the 3-D Secure authentication details from a third-party, in which case they should provide them as part of a standard request. If the Gateway receives valid third-party authentication details, then it will use those and not attempt to contact the 3-D Secure system itself.

1.4 Request Fields

1.4.1 Initial Request

These fields should be sent in addition to basic request fields in section 2.1.

Field Name	Mandatory?	Description
merchantName	No[1]	Merchant name to use on 3DS form.
merchantWebsite	No[1]	Merchant website to use on 3DS form.
threeDSRequired	No[2]	Is 3DS required for this transaction? Possible values are: **N** – 3DS is not required. **Y** – Abort if 3DS is not enabled.
threeDSCheckPref	No[1]	List of **threeDSCheck** response values that are to be accepted, any other value will cause the transaction to be declined. Value is a comma separated list containing one or more of the following values: **'not known', 'not checked', ' not authenticated', 'attempted authentication', 'authenticated'**.

[1] If the value is not supplied than the default account preferences will be used.
[2] The default value is **Y** if 3-D Secure is enabled on the Merchant Account, otherwise **N**

1.4.1 Continuation Request

These fields may be sent alone.

Field Name	Mandatory?	Description
threeDSMD	Yes	The value of the **threeDSMD** field in the initial Gateway response.
threeDSPaRes	Yes	The value of the **PaRes** field POSTed back from the Access Control Sever (ACS)

Note: It is only necessary to send the **threeDSMD** and the **threeDSPaRes** in the continuation request as the **threeDSMD** will identify the Merchant Account and initial request. The message does not need to be signed. However you can send any of the normal request fields to modify or supplement the initial request. Any card details and transaction amount sent in the second request must match those used in the first request, or the second request will fail with a **responseCode** of **64442 (REQUEST MISMATCH)**.

1.4.1 External Authentication Request

These fields should be sent in addition to basic request fields from section 2.1.

Field Name	Mandatory?	Description
threeDSEnrolled	If 3DS enabled	The 3DS enrolment status for the credit card. Refer to appendix A-3 for details. Possible values are: **Y** – Enrolled. **N** - Not Enrolled. **U** - Unable to Verify.
threeDSAuthenticated	If 3DS enrolled	The 3DS authentication status for the credit card. Refer to appendix A-3 for details. Possible values are: **Y** - Authentication Successful. **N** - Not Authenticated. **U** - Unable to Authenticate. **A** - Attempted Authentication.
threeDSXID	If 3DS authenticated	The unique identifier for the transaction in the 3DS system.
threeDSECI	If 3DS authenticated	The Electronic Commerce Indicator (ECI).
threeDSCAVV	If 3DS authenticated	The Cardholder Authentication Verification Value (CAVV).

Note: If 3-D Secure is not enabled for the Merchant Account then any 3-D Secure authentication fields sent in the request are ignored and the transaction is processed as normal without 3-D Secure.

1.5 Response Fields

1.5.1 Initial Response

These fields will be returned in addition to the request fields from section 5.4.1 and the basic response fields in section 0.

Field Name	Returned?	Description
threeDSEnabled	Always	Is 3DS enabled for this Merchant Account? Possible values are: **N** – Merchant Account is not enabled. **Y** – Merchant Account is enabled.
threeDSXID	If 3DS enabled	The unique identifier for the transaction in the 3DS system.
threeDSVETimestamp	If 3DS enabled	The time the card was checked for 3DS enrolment.
threeDSEnrolled	If 3DS enabled	The 3DS enrolment status for the credit card. Refer to appendix A-3 for details. Possible values are: **Y** – Enrolled. **N** - Not Enrolled. **U** - Unable to Verify. **E** - Error Verifying Enrolment.
threeDSMS	If 3DS enabled	Value to return in the continuation request. Can be sent to the Access Control Server (ACS) in its MD field or stored locally by your server.
threeDSACSURL	If 3DS enrolled	The URL of the Access Control Server (ACS) to which the Payer Authentication Request (PaReq) should be sent.
threeDSPaReq	If 3DS enrolled	Payer Authentication Request (PaReq) that is sent to the Access Control Server (ACS) in order to verify the 3DS status of the credit card.

1.5.2 Continuation Response

These fields will be returned in addition to the request fields from section 5.4.1, the initial response fields in section 5.5.1 and the basic response fields in section 0.

Field Name	Returned?	Description
threeDSPaRes	If 3DS enrolled	Payer Authentication Response (PaRes) that is returned from the Access Control Server (ACS) determining the 3DS status of the credit card.
threeDSCATimestamp	If 3DS enrolled	The time the card was checked for 3DS authentication.
threeDSAuthenticated	If 3DS enrolled	The 3DS authentication status for the credit card. Refer to appendix A-3 for details. Possible values are: **Y** - Authentication Successful. **N** - Not Authenticated. **U** - Unable to Authenticate. **A** - Attempted Authentication. **E** - Error Checking Authentication..
threeDSECI	If 3DS authenticated	This contains a two digit Electronic Commerce Indicator (ECI) value, which is to be submitted in a credit card authorisation message. This value indicates to the processor that the Customer data in the authorisation message has been authenticated. The data contained within this property is only valid if the **threeDSAuthenticated** value is **Y** or **A**.
threeDSCAVV	If 3DS authenticated	This contains a 28-byte Base-64 encoded Cardholder Authentication Verification Value (CAVV). The data contained within this property is only valid if the **threeDSAuthenticated** value is **Y** or **A**.
threeDSCAVVAlgorithm	If 3DS authenticated	This contains the one digit value which indicates the algorithm used by the Access Control Server (ACS) to generate the CAVV. The data contained within this property is only valid if the **threeDSAuthenticated** value is **Y** or **A**.

Field Name	Returned?	Description
threeDSErrorCode	If 3DS error	Any error response code returned by the Access Control Server (ACS) should there be an error in determining the card's 3DS status.
threeDSErrorDescription	If 3DS error	Any error response description returned by the Access Control Server (ACS) should there be an error in determining the card's 3DS status.

1.5.3 External Authentication Response

These fields will be returned in addition to the request fields from section 5.4.3 and the basic response fields in section 0.

Field Name	Returned?	Description
threeDSEnabled	Always	Is 3DS enabled for this Merchant Account? Possible values are: **N** – Merchant Account is not enabled. **Y** – Merchant Account is enabled.

Note: If 3-D Secure is not enabled for the Merchant Account then any 3-D Secure authentication fields sent in the request are ignored and the transaction is processed as normal without 3-D Secure.

1 VISA MCC6012 Merchants

1.1 Background

Following changes implemented by VISA, any UK business falling under merchant category code 6012 must provide additional details with any transaction that is processed through their account. This mainly applies to financial institutions.

According to Visa, the additional rules were brought in to protect consumers and prevent fraud. The Acquirer will inform you if they need to send this information.

1.1.1 Requirements

This section only applies to transactions that:

- Involve a Merchant with a MCC 6012 category code.
- Use VISA
- Process a UK domestic payment.

If any of the above three criteria do not apply, then no additional data need be supplied in the transaction.

1.1.2 Additional fields/information

Merchants assigned the code MCC 6012 must collect the following data for the primary recipient for each UK domestic VISA transaction:

- Unique account identifier for the loan or outstanding balance funded. For example, the loan account number or the PAN (Primary Account Number) if it is a credit card balance.
- Last name (family name)
- Date of Birth (D.O.B)
- Postcode

Primary recipients are the entities (people or organisations) that have a direct relationship with the financial institution. Also, these primary recipients have agreed to the terms and conditions of the financial institution.

1.7 Request Fields

To comply with the rules, an MCC6012 Merchant must send these additional fields:

Field Name	Mandatory?	Description
merchantCategoryCode	Yes[1]	Merchant's VISA MCC (should be 6012).
receiverName	Yes	Surname only - up to 6 letters allowed.
receiverAccountNo	Yes	Account number. If a PAN is supplied the only the first 6 and last 4 digits will be used.
receiverDateOfBirth	Yes	Primary recipient's date of birth. **ISO Date Format: YYYY-MM-DD.**
receiverPostcode	Yes	Primary recipient's postcode. (Only the district is required but full postcodes are accepted, therefore 'W12 8QT' or just 'W12' are acceptable values).

[1] Only required if the Merchants Category Code is not configured on their gateway account.

1 Billing Descriptor

1.1 Background

The Billing Descriptor is how the Merchant's details appear on the cardholder's statement. It is set up with the Acquirer when the Merchant Account is opened. It is used by the cardholder to identify who a payment was made to on a particular transaction.

Selecting a clear Billing Descriptor is important for a Merchant to avoid a chargeback when the cardholder does not recognise the name on the transaction.

1.1.1 Static Descriptor

The Static Descriptor is the descriptor agreed between the Merchant and Acquirer when the Merchant Account is opened. The descriptor used is typically the Merchant's trading name, location and contact phone number.

1.1.2 Dynamic Descriptor

The Dynamic Descriptor is a descriptor sent with the transaction that includes details on the goods purchased or service provided, this is often used by large companies that provide many services and where the brand of the service is more familiar than the company name. The Dynamic Descriptor usually replaces any Static Descriptor on a per transaction basis.

Not all Acquirers accept Dynamic Descriptors and for those that do the format required varies. Often the Merchant's name is shortened to three (3) letter, followed by an asterisk (*), followed by a short description of the service or product that the business provides. This field typically has a limit of twenty-five (25) characters including the phone number

For more information on whether your Acquirer allows Dynamic Descriptor and the format they should be sent in please contact your support representative.

1.2 Request Fields

The Dynamic Descriptor is built using one or more of the following narrative fields.

Field Name	Mandatory?	Description
`statementNarrative1`	No	Merchant's name.
`statementNarrative2`	No	Product, service or other descriptive info.

1 Receipts & Notifications

1.1 Background

The Gateway can be configured to automatically email transaction receipts to the Customer and notifications to the Merchant. The Gateway is also integrated into the eReceipts™ system which stores Customer receipts for access online.

1.1.1 Customer Email Receipts

The Customer can be automatically emailed a transaction receipt each time a transaction is processed by the Gateway. Receipts are sent at the time the transaction is authorised and only for transactions where the Acquirer has approved the authorisation. Receipts are not sent for declined or referred authorisations or aborted transactions.

This functionality is enabled globally on a per Merchant Account basis using the Merchant Management System (MMS). This global setting can also be overridden per transaction if required using the **`customerReceiptsRequired`** field.

Customer receipts require the Customer to provide their email address; if no email address is sent in the **`customerEmail`** field then no receipt will be sent.

1.1.2 Merchant Email Notifications

The Merchant can be automatically emailed a transaction notification each time a transaction is processed by the Gateway. Notifications are sent at the time the transaction is authorised and only for transactions where the Acquirer approved, declined or referred the authorisation. Notifications are not sent for aborted transactions.

This functionality is enabled globally on a per Merchant Account basis using the Merchant Management System (MMS). This global setting can also be overridden per transaction if required using the **`notifyEmailRequired`** field.

1.1.3 Customer Online Receipts

The Gateway is integrated with the eReceipts™ system run by Paperless Receipts Ltd. This system is used by many high street retailers and allows a Merchant to capture data allowing a far deeper understanding and insight into their Customers' shopping habits. Receipt information is sent to eReceipts™ at the time the transaction is authorised and only for transactions where the Acquirer has approved the authorisation. Receipt information is not sent for declined or referred authorisations or aborted transactions.

This functionality is enabled globally on a per Merchant Account basis using the Merchant Management System (MMS). This global setting can also be overridden per transaction if required using the **`eReceiptsRequired`** field.

Merchant must supply a unique Customer reference (using the **`eReceiptsCustomerRef`** field) or, alternatively, the use the Customer's email address (using the **`customerEmail`** field) to identify the Customer in the eReceipts™ system.

If purchase item data is sent in a transaction, then this will be used to build an itemised electronic receipt. For more information regarding purchase data please refer to section 8.3, for information on which fields are used to build the electronic receipt refer to section 8.2 below.

1.2 Request Fields

1.2.1 General Fields

Field Name	Mandatory?	Description
`customerReceiptsReqired`	No[1]	Send a Customer receipt if possible. Possible values are: **N** – Don't send a receipt. **Y** – Send if Customer's email provided.
`customerEmail`	No	Customer's email address.
`notifyEmailRequired`	No[1]	Send a notification email if possible. Possible values are: **N** – Don't send a notification. **Y** – Send if notification email provided.
`notifyEmail`	No[1]	Merchant's notification email address.
`eReceiptsRequired`	No[1]	Send receipt data to eReceipts™ system. Possible values are: **N** – Don't send data. **Y** – Send data if API details provided.
`eReceiptsStoreID`	No[1]	eReceipts™ store identifier.
`eReceiptsCustomerRef`	No[2]	eReceipts™ Customer reference.
`eReceiptsApiKey`	No[1]	eReceipts™ API key.
`eReceiptsApiSecret`	No[1]	eReceipts™ API secret.
`eReceiptsReceiptRef`	No	eReceipts™ receipt reference.
`eReceiptsReceiptData`	No[3]	Complete eReceipts™ data

[1] Overrides any global setting configured via the Merchant Management System (MMS).
[2] Required if eReceipts™ is required and no customerEmail is sent.
[3] Allows complete eReceipts™ data to be sent rather than constructing it from the transaction.

1.2.1 eReceipts™ Itemised Receipt Data

Field Name	Mandatory?	Description
grossAmount	No	Total gross amount of sale.
netAmount	No	Total net amount of sale.
taxAmount	No	Total tax amount of sale.
taxRate	No	Total tax rate (percentage).
discountAmount	No	Total discount amount of sale.
discountReason	No	Reason for above discount.
item*XX*Description	No	Description of *XX*th item purchased.
item*XX*Quantity	No	Quantity of *XX*th item purchased.
item*XX*GrossAmount	No	Gross amount for *XX*th item purchased.
item*XX*TaxAmount	No	Tax amount for *XX*th item purchased.
item*XX*TaxRate	No	Total tax rate for *XX*th item purchased.
item*XX*DiscountAmount	No	Total discount for *XX*th item purchased.
item*XX*DiscountReason	No	Discount reason for *XX*th item purchased.
item*XX*ProductCode	No	Product code for *XX*th item purchased.
item*XX*CommodityCode	No	Commodity code for *XX*th item purchased.
item*XX*UnitOfMeasure	No	Unit of measure for *XX*th item purchased.
item*XX*UnitAmount	No	Unit amount for *XX*th item purchased.
items	No[1]	Nested array of line items.

[1] Used as an alternative to **itemXXField** format, both formats can not be sent together.

Note: no attempt is made to check that any gross, net and tax amounts are correct with respect to each other. It is the sender's responsibility to ensure alternative amount formats are correct.

1.1 Response Fields

The request fields for the required receipts and notifications are returned along with the appropriate fields from the following.

Field Name	Returned?	Description
`customerReceiptsResponseCode`	If required	Result of sending email to Customer. Refer to appendix A-1 for details.
`customerReceiptsResponseMessage`	If required	Description of above response code.
`notifyEmailResponseCode`	If required	Result of sending email to Merchant. Refer to appendix A-1 for details.
`notifyEmailResponseMessage`	If required	Description of above response code.
`eReceiptsEnabled`	If required	Is eReceipts™ enabled for this Merchant Account? Possible values are: **N** – Merchant Account is not enabled. **Y** – Merchant Account is enabled.
`eReceiptsStoreID`	If required[1]	Merchant's eReceipts™ store identifier.
`eReceiptsReceiptRef`	If required	Unique eReceipt™ reference.
`eReceiptsResponseCode`	If required	Result of sending details to eReceipts™. Refer to appendix A-1 for details.
`eReceiptsResponseMessage`	If required	Description of above response code.

[1] Either the value sent in the request or that calculated from the default account preferences

1 Purchase Data

1.1 Background

The Gateway can be sent advance purchase information with each transaction where required.

The Gateway provides a number of fields which you can use to store advanced purchase information about the transaction including details on individual items purchased etc. These fields are only sent to the Acquirer if needed. The stored data can be obtained by sending a QUERY request.

The details may also be used for advanced purposes such as displaying shopping cart information on the MasterPass™ checkout or sending full receipt details to the eReceipts™ system.

1.1.1 American Express Purchases

Purchases using American Express cards will send a subset of this information to the card scheme as appropriate.

With American Express you can provide tax ***or*** discount reason (but not both). If **`taxAmount`** is provided then **`taxReason`** is used, if **`discountAmount`** is provided then **`discountAmount`** is used. If both are provided then **`taxReason`** is used.

Only the first six line item details are sent to American Express and then only the **`itemXXDescription`**, **`itemXXQuantity`** and **`itemXXGrossAmount`** fields are sent.

1.1.2 Purchase Orders

These fields along with other advanced fields as detailed in section 12 can be used to send full information relating to a purchase order and related invoice indicating types, quantities and agreed prices for products or services. Details on the supplier, shipping, delivery etc. can also be included.

At present this information is not sent to the Acquirer but future enhancements to the Gateway may include sending such information as Level 2 or 3 Purchasing data as defined by the relevant card schemes.

1.2 Request Fields

Field Name	Mandatory?	Description
grossAmount	No	Total gross amount of sale.
netAmount	No	Total net amount of sale.
taxRate	No	Total tax rate (percentage).
taxAmount	No[1]	Total tax amount of sale.
taxReason	No[1]	Reason for above tax (ie VAT).
discountAmount	No[1]	Total discount amount of sale.
discountReason	No[1]	Reason for above discount.
item*XX*Amount[2]	No	Amount for *XX*th item purchased.
item*XX*Description[2]	No	Description of *XX*th item purchased.
item*XX*Quantity[2]	No	Quantity of *XX*th item purchased.
item*XX*GrossAmount[2]	No	Gross amount for *XX*th item purchased.
item*XX*NetAmount[2]	No	Net amount for *XX*th item purchased.
item*XX*TaxAmount[2]	No	Tax amount for *XX*th item purchased.
item*XX*TaxRate[2]	No	Total tax rate for *XX*th item purchased.
item*XX*TaxReason[2]	No	Tax reason for *XX*th item purchased.
item*XX*DiscountAmount[2]	No	Total discount for *XX*th item purchased.
item*XX*DiscountReason[2]	No	Discount reason for *XX*th item purchased.
item*XX*ProductCode[2]	No	Product code for *XX*th item purchased.
item*XX*CommodityCode[2]	No	Commodity code for *XX*th item purchased.
item*XX*UnitOfMeasure[2]	No	Unit of measure for *XX*th item purchased.
item*XX*UnitAmount[2]	No	Unit amount for *XX*th item purchased.
items	No[3]	Nested array of line items.

[1] Amex/Diners require either tax or discount not both
[2] XX is a number between 1 and 99
[3] Used as an alternative to **itemXXField** format, both formats can not be sent together.

Note: no attempt is made to check that any gross, net and tax amounts are correct with respect to each other. It is the sender's responsibility to ensure alternative amount formats are correct.

Line item fields can either be sent ‘flat’ using field names containing the item row number as a sequential number from 1 to 99 or using nested arrays of the form **`items[XX][field]`** where ***`XX`*** is the row number from 1 to 99 and ***`field`*** is the field name from the above table without the **`item`*`XX`*** prefix and starting with a lowercase first letter. For example, the tax rate for item 5 can either be sent as **`item5TaxRate`** or as **`items[5][taxRate]`**. The two formats should not be mixed. If a request field of **`items`** is seen then the ‘flat’ fields are ignored.

1 Recurring Transaction Agreements

1.1 Background

A Recurring Transaction Agreement (RTA) is used to request that the Gateway repeat payments on behalf of the Merchant using pre agreed amounts and schedule.

An RTA can be easily and quickly configured using the Merchant Management System (MMS) but can also be setup while performing the initial request by including the following integration request fields. The RTA is only setup in the transaction results in a success payment authorisation.

The initial transaction should be either a normal transaction or the initial transaction in a Continuous Payment Authority agreement. This will dictate whether the subsequent repeat transactions are taken as ‘Card on File’ or ‘Continuous Authority’ transactions. Refer to Appendix 14.7A-15 for more information on the different types of repeat or recurring transactions.

1.2 Request Fields

Field Name	Mandatory?	Description
rtName	No	Free format short name for the agreement.
rtDescription	No	Free format longer description for the agreement.
rtPolicyRef	No	Merchant Reference (MPRN).
rtAgreementType	No	Recurring transaction agreement type. When provided the initial transaction will be marked as the first in a Continuous Payment Authority (CPA) agreement and subsequent scheduled repeat transactions will be made as part of that CPA. If not provided then the initial transaction will be a standard transaction and the subsequent scheduled repeat transactions will be taken as ad-hoc Card On File transactions. Possible values are: <not provided> - use Card On File **recurring** – use a recurring type CPA. **instalment** – use an instalment type CPA
rtUnique	No	Unique id for recurring transactions, will be append with transaction count (defaults to **transactionUnique**).
rtMerchantID	No	Merchant ID to use for the recurring transactions (defaults to **merchantID**).
rtStartDate	No	Start date of agreement (default to date request received).
rtInitialDate	No	Date of initial payment (defaults to rtStartDate). Format: YYYY-MM-DD HH:MM:SS
rtInitialAmount	No	Amount of initial payment (defaults to **rtCycleAmount**).
rtFinalDate	No	Date of final payment. Format: YYYY-MM-DD HH:MM:SS.
rtFinalAmount	No	Amount of final payment (defaults to **rtCycleAmount**).
rtCycleAmount	No	Amount per cycle (defaults to **amount**).
rtCycleDuration	Yes	Cycle duration.

rtCycleDurationUnit	Yes	Cycle duration unit. One of the following values: **day**, **week**, **month** or **year**.
rtCycleCount	No	Number of cycles to repeat (defaults to repeat forever).
rtMerchantData	No	Free format Merchant data field

1.3 Response Fields

Field Name	Returned?	Description
`rtID`	Always	Recurring Transaction Agreement ID.
`rtResponseCode`	Always	Result of setting up RT Agreement. Refer to appendix A-1 for details.
`rtResponseMessage`	Always	Description of above response code.

1 Duplicate Transaction Checking

1.1 Background

Duplicate transaction checking prevents transaction requests from accidentally processing more than once. This can happen if a Customer refreshes your checkout page or clicks a button that issues a new transaction request. While duplicate checking can help prevent repeat transactions from going through, we recommend talking with your developers to see if changes can be made to your form to reduce the likelihood of this occurring (e.g. disabling the Submit button after it's clicked).

1.2 Implementation

To help prevent duplicate transactions each transaction can specify a time window during which during which previous transactions will be check to see if they could be possible duplicates.

This time window is specified using the **`duplicateDelay`** field. The value for this field can range from 0 to 9999 seconds (approx 2 ¾ hours).

If the transaction request does not include the **`duplicateDelay`** field or specifies a value of zero, then a default delay of 300 seconds (5 minutes) is used.

The following fields are used in transaction comparison and must be the same for a transaction to be regarded as a duplicate;

- **`merchantID`**
- **`action`**
- **`type`**
- **`amount`**
- **`transactionUnique`**
- **`currencyCode`**
- **`xref`** (if provided in lieu of card details)
- **`cardNumber`** (may be specified indirectly via cross reference)

If a transaction is regarded as being a duplicate, then a **`responseCode`** of **65554 (REQUEST DUPLICATE**) will be returned.

1.3 Request Fields

Field Name	Mandatory?	Description
`duplicateDelay`	No	Duplicate transaction time window in seconds. **Numeric value between 0 and 9999.**

1 Custom Data

You may send arbitrary data with the request by appending extra fields, which will be returned in the response unmodified. These extra fields are merely 'echoed' back and not stored by the Payment Gateway.

Caution should be made to ensure that any extra fields do not match any currently documented fields or possible future fields; one way to do this is to prefix the field names with a value unique to the Merchant.

You can also use the **`merchantData`** field to store custom data with the transaction. This stored data can then be retrieved at a later date using a QUERY request. Associative data can be serialised using the notation **`merchantData[name]=value`**.

1.6 Request Fields

Field Name	Mandatory?	Description
`merchantData`	No	Arbitrary data to be stored along with this transaction.

1 Advanced Integration Fields

The Gateway provides a number of fields that you can use to store information about the transaction. These fields are only sent to the Acquirer if needed. The stored data can be obtained by sending a QUERY request.

1.1 Customer Request Fields

These fields can be used to store details about the Customer and any relationship between the Customer and Merchant such as any purchase order raised etc.

If AVS checks are in use, then the Customer and cardholder are assumed to be the same person and the address and postcode fields are taken as being the registered billing address of the card.

Field Name	Mandatory?	Description
customerName	No	Cardholder's name.
customerCompany	No	Cardholder's company (if applicable)
customerAddress	No[1]	Cardholder's address.
customerPostcode	No[1]	Cardholder's postcode.
customerTown	No	Cardholder's town/city.
customerCounty	No	Cardholder's county/province.
customerCountryCode	No	Cardholder's country. ISO-3166 alpha or numeric code.
customerPhone	No	Cardholder's phone number
customerMobile	No	Cardholder's mobile phone number.
customerFax	No	Cardholder's fax number.
customerEmail	No	Cardholder's email address.
customerOrderRef	No	Customer's reference for this order. (Purchase Order Reference)
customerMerchantRef	No	Customer's reference for the Merchant.
customerTaxRef	No	Customer's tax reference number.

[1] Mandatory if AVS checking required

1.1 Merchant Request Fields

These fields can be used to store details about the Merchant and any relationship between the Merchant and Customer such as any invoice reference etc.

Field Name	Mandatory?	Description/Value
merchantName	No	Merchant's contact name.
merchantCompany	No	Merchant's company name.
merchantAddress	No	Merchant's contact address.
merchantTown	No	Merchant's contact town/city.
merchantCounty	No	Merchant's contact county.
merchantPostcode	No	Merchant's contact postcode.
merchantCountryCode	No	Merchant's contact country. **Valid ISO-3166 alpha or numeric code.**
merchantPhone	No	Merchant's phone.
merchantMobile	No	Merchant's mobile phone number.
merchantFax	No	Merchant's fax number.
merchantEmail	No	Merchant's email address.
merchantWebsite	No	Merchant's website.
merchantOrderRef	No	Merchant's reference for this order. (Invoice/Sales Reference)
merchantCustomerRef	No	Merchant's reference for the Customer.
merchantTaxRef	No	Merchant's tax reference number.
merchantOriginalOrderRef	No	Reference to a back order.
merchantCategoryCode	No	Scheme assigned Merchant Category Code (MCC).

1.2 Supplier Request Fields

These fields can be used to store details about the Supplier address. This is where any purchased goods are being supplied from, if different to the Merchant's address.

Field Name	Mandatory?	Description/Value
supplierName	No	Supplier's contact name.
supplierCompany	No	Supplier's company name.
supplierAddress	No	Supplier's contact address.
supplierTown	No	Supplier's contact town/city.
supplierCounty	No	Supplier's contact county.
supplierPostcode	No	Supplier's contact postcode.
supplierCountryCode	No	Supplier's contact country. **Valid ISO-3166 alpha or numeric code.**
supplierPhone	No	Supplier's phone.
supplierMobile	No	Supplier's mobile phone number.
supplierFax	No	Supplier's fax number.
supplierEmail	No	Supplier's email address.

1.3 Delivery Request Fields

These fields can be used to store details about the delivery address. This is where any purchased goods are being delivered to if different to the Customer's address.

Field Name	Mandatory?	Description/Value
deliveryName	No	Name of person receiving the delivery.
deliveryCompany	No	Name of company receiving the delivery.
deliveryAddress	No	Delivery address.
deliveryTown	No	Delivery town/city.
deliveryCounty	No	Delivery county.
deliveryPostcode	No	Delivery postcode.
deliveryCountryCode	No	Delivery country. **Valid ISO-3166 alpha or numeric code.**
deliveryPhone	No	Phone number of delivery location.
deliveryMobile	No	Mobile phone number of delivery location.
deliveryFax	No	Fax number of delivery location.
deliveryEmail	No	Delivery email address.

1.4 Receiver Request Fields

These fields can be used to store details about the recipient of the purchased goods where different to the Customer's and Delivery details. It is most commonly used by Financial Intuitions (MCC 6012 Merchants) who need to record the primary recipient of a loan etc.

Field Name	Mandatory?	Description/Value
receiverName	No	Receiver's contact name.
receiverCompany	No	Receiver's company name.
receiverAddress	No	Receiver's contact address.
receiverTown	No	Receiver's contact town/city.
receiverCounty	No	Receiver's contact county.
receiverPostcode	No	Receiver's contact postcode.
receiverCountryCode	No	Receiver's contact country. **Valid ISO-3166 alpha or numeric code.**
receiverPhone	No	Receiver's phone.
receiverMobile	No	Receiver's mobile phone number.
receiverFax	No	Receiver's fax number.
receiverEmail	No	Receiver's email address.
receiverAccountNo	No	Receiver's account number.
receiverDateOfBirth	No	Receiver's date of birth.

1.5 Shipping Request Fields

These fields can be used to store details about the shipping method and costs.

Field Name	Mandatory?	Description/Value
shippingTrackingRef	No	Shipping tracking reference.
shippingMethod	No	Shipping method (eg. Courier, Post, etc.).
shippingAmount	No	Cost of shipping.
shippingGrossAmount	No	Gross cost of shipping.
shippingNetAmount	No	Net cost of shipping.
shippingTaxRate	No	Tax rate as percentage to 2 decimal places.
shippingTaxAmount	No	Tax cost of shipping.
shippingTaxReason	No	Tax reason (ie. VAT).
shippingDiscountAmount	No	Discount on shipping.
shippingDiscountReason	No	Reason for discount.

Note: no attempt is made to check that any gross, net and tax amounts are correct with respect to each other. It is the sender's responsibility to ensure alternative amount formats are correct.

1 PayPal Transactions

1.1 Background

PayPal is an additional payment method that is available to all Merchants using the Payment Gateway that have a PayPal account.

It allows the Merchant to offer payment via PayPal in addition to normal card payments.

PayPal transactions will appear in the Merchant Management System (MMS) alongside any card payments and can be captured, cancelled and refunded in the same way as card payments.

PayPal transactions can also be used for recurring billing but require the Merchant to indicate in the initial transaction that it will be basis for recurring billing and a billing agreement will be entered into between your customer and PayPal when they agree to the payment.

PayPal transactions cannot be used for ad-hoc ‘card-on-file’ repeat transactions unless a billing agreement has been set up.

For more information on how to accept PayPal transactions please contact your support representative.

1.2 Benefits & Limitations

1.2.1 Benefits

- Instant: The PayPal transaction information is available immediately and returned as part of the transaction

- Flexible: Adding PayPal gives your customers the flexibility of paying using their PayPal account when this is more suitable to them than using a traditional credit or debit card.

- Express Checkout: The in-context PayPal Express Checkout helps improve conversion rates with an easier way to pay without customers leaving your website.

- No extra cost: There are no extra costs to add PayPal to your gateway account however you will still be liable for the PayPal transaction fees.

- Easy management: The PayPal transactions are controlled within the Merchant Management System (MMS).

1.2.2 Limitations

- You will need a PayPal account in order to process PayPal transactions as well as a normal Acquirer account to process card transactions.

- Ad-hoc repeat ‘card on file’ transactions are not supported unless part of a prearranged PayPal billing agreement.

- Independent refunds which are not tied to a previous PayPal sale transaction are not supported without prior agreement with PayPal.

- The PayPal checkout cannot be opened from within a browser IFRAME and so care must be taken to ensure that any PayPal checkout button is not placed within such an IFRAME.

1.3 Implementation

To use PayPal you will be supplied with a separate PayPal Merchant account which can be grouped with your normal card Merchant account using the account mapping facility as documented in Appendix A-6. This allows transactions to be sent using your main Merchant Account and then routed automatically to the PayPal Merchant Account in the same mapping group.

Initial PayPal transactions require you to display the PayPal Checkout to your customer as part of the transaction flow. In this respect they work very must like 3-D Secure transactions and need to be done in two stages with the Checkout being displayed between the stages. Like 3-D Secure they can also be optionally done in three stages allowing you to display an order confirmation after the Checkout and before authorising the transaction. Unlike 3-D Secure you can change the amount at this stage to allow for shipping costs once you know the confirmed delivery address the customer selected as part of the PayPal Checkout.

1.3.1 Initial Request (Checkout Preparation)

To request a transaction be processed via PayPal the request must contain a **`paymentMethod`** of 'paypal' and a **`checkoutRedirectURL`** containing the URL of a page on the Merchant's server to return to when the Checkout is closed. In addition, you may send **`checkoutOptions`** to customise the Checkout experience. When the Gateway receives these two fields, assuming there are no other errors with the request, it will attempt to find a suitable PayPal Merchant Account in the current account mapping group.

If the Gateway is unable to find a suitable account, then the transaction will be aborted and it will response with a **`responseCode`** of **66364 (INVALID PAYMENTMETHOD)**.

Otherwise the Gateway will respond with a **`responseCode`** of **65826 (CHECKOUT REQUIRED)** and included in the response will be a **`checkoutURL`** field containing the URL required to load Checkout and a **`checkoutRequest`** containing any data required to be sent to the Checkout. The response will also contain a unique **`checkoutRef`** which needs to be echoed back in the continuation requests.

At this point the Merchant's server needs to either redirect to customer's browser to the provided **`checkoutURL`**.

The **`checkoutURL`** can also be used in conjunction with the PayPal In-Context JavaScript code to implement an In-context checkout which allows the Merchants website to remain visible in the background. Further details on how to use the In-Context checkout are provided in the PayPal guide at

https://developer.paypal.com/docs/classic/express-checkout/in-context/enable_in_context_checkout/.

1.3.2 Continuation Request (Checkout Details & Authorise)

On completion of the PayPal Checkout it will redirect the customer's browser to the **`checkoutRedirectURL`** provided including a **`token`** and **`status`** URL parameters. If the checkout was successful, the status will be 'success' alternatively if the checkout was cancelled the status will be 'cancel'. The received redirect request parameters inclusive of these **`token`** and **`status`** parameters should then be sent to the Gateway in the **`checkoutResponse`** fields of a new request. The **`checkoutResponse`** field can be sent either as the original URL query string received or as an array of the decoded query parameters. This new request will load the checkout details including any delivery address if required and send the transaction to PayPal for authorisation, returning the result as per a normal authorisation transaction. The new request must contain the **`checkoutRef`** received in the initial response.

1.3.3 Separate Checkout Details & Authorisation Requests

You can choose to obtain the Checkout details before actually sending the transaction for authorisation by sending the checkoutOnly field in the above continuation request. If this field is sent with a value of 'Y' then the Gateway will load the checkout details and then return them to the Merchant without sending the request for authorisation. The Merchant can then display them and/or adjust the amount, for example, according to delivery charges dependant on the received delivery address. The Merchant should then send a new request containing the **`checkoutRef`** received to continue the transaction and authorise it.

Note: this stage can be repeated multiple times by including the **`checkoutOnly`** field with a value of 'Y' each time. To complete the transaction, the final request must not contain the **`checkoutOnly`** field or it must not have a value of 'Y'.

1.4 Request Fields

1.4.1 Initial Request

These fields should be sent in addition to basic request fields in section 2.1 excluding any card details.

Field Name	Mandatory?	Description
paymentMethod	Yes	Must contain the value 'paypal' in lower case letters only.
checkoutRedirectURL	Yes	URL on Merchant's server to return to when the PayPal Checkout is closed.
checkoutOptions	No[1]	Associative array or URL encoded array of options used to customise the PayPal Checkout. Refer to section 14.4.3 for values.

1.4.1 Continuation Request

These fields may be sent alone.

Field Name	Mandatory?	Description
checkoutRef	Yes	Unique reference return in the initial response
checkoutResponse	Yes	The GET and or POST data received by the **checkoutRedirectURL** page
checkoutOnly	No	Pass **Y** to complete the processing as far as the next checkout stage and then return with the loaded checkout details.

Note: It is only necessary to send the **checkoutRef** and the **checkoutResponse** in the continuation request as the **checkoutRef** will identify the Merchant Account and initial request. The message does not need to be signed. You can send any of the normal request fields to modify or supplement the initial request, however in this case the request should be signed. The **checkoutRedirectURL** and **checkoutOptions** fields sent in the initial request can not be modified and any sent in the second request must match those used in the first request, or the second request will fail with a **responseCode** of **64442 (REQUEST MISMATCH)**.

1.4.1 Checkout Options

The following options may be sent in the **`checkoutOptions`** field to customise the PayPal checkout. The field may be sent as a URL encoded string or an array of key/value pairs.

Field Name	Mandatory?	Description
`inContext`	No	Use the in-context PayPal checkout rather than the full screen checkout when possible. Possible values are: **0** – use the full screen checkout. **1** – use the in-context checkout if possible.
`userAction`	No	Determines whether buyers complete their purchases on PayPal or on your website. Possible values are: **commit** – sets the submit button text to 'Pay Now' on the PayPal checkout. This text lets buyers know that they complete their purchases if they click the button. **continue**– sets the submit button text to 'Continue' on the PayPal checkout. This text lets buyers know that they will return to the Merchants cart to complete their purchases if they click the button.
`maxAmount`	No[1]	The expected maximum total amount of the order, including shipping and taxes.
`reqConfirmShipping`	No	Determines whether the shipping address on file with PayPal must a confirmed address. Possible values are: **0** – does not need to be confirmed **1** – must be confirmed
`noShipping`	No	Determines whether PayPal displays shipping address. Possible values are: **0** – display the shipping address **1** – do not display shipping address and remove shipping information **2** – If no `deliveryXXX` fields passed PayPal obtains them from the buyer's account profile.
`addrOverride`	No	Determines whether the PayPal checkout displays the shipping address sent using the `deliveryXXX` fields and not the shipping address on file with PayPal for this buyer. Displaying the PayPal street address on file does not allow the buyer to edit that address.

Field Name	Mandatory?	Description
		Possible values are: **0** – PayPal should not display the address. **1** – PayPal should display the address.
localeCode	No	Locale of the pages displayed by PayPal during Express Checkout. It is either a two-letter country code or five-character locale code supported by PayPal.
allowNote	No	Enables the buyer to enter a note to the merchant on the PayPal page during checkout. The note is returned in the `checkoutDetails` response field.
pageStyle	No	Name of the Custom Payment Page Style used for the PayPal checkout. It is the same name as the Page Style Name used when adding styles in the PayPal Account.
payflowColor	No	The HTML hex colour code for the PayPal checkout's background colour. By default, the colour is white (FFFFFF).
cardBorderColor	No	The HTML hex colour code for the PayPal checkout's principle identifying colour. The colour will be blended to white in a gradient fill that borders the cart review area.
hdrImg	No	URL for the image you want to appear at the top left of the payment page. The image has a maximum size of 750 pixels wide by 90 pixels high. PayPal requires that you provide an image that is stored on a secure (https) server. If you do not specify an image, the business name displays.
logoImg	No	A URL to your logo image. Use a valid graphics format, such as .gif, .jpg, or .png. Limit the image to 190 pixels wide by 60 pixels high. PayPal crops images that are larger. PayPal places your logo image at the top of the cart review area.
landingPage	No	Type of PayPal checkout to display. Possible values are: **Billing** – Non-PayPal account **Login** – PayPal account login
channelType	No	Type of channel. Possible values are: **Merchant** – Non-auction seller **eBayItem** – eBay auction

Field Name	Mandatory?	Description
solutionType	No	Type of checkout flow. Possible values are: **Sole** – Buyer does not need to create a PayPal account to check out. This is referred to as PayPal Account Optional. **Mark** – Buyer must have a PayPal account to check out.
totalType	No	Type declaration for the label to be displayed in MiniCart for UX. Possible values are: **Total** **EstimatedTotal**
brandName	No	A label that overrides the business name in the PayPal account on the PayPal checkout.
customerServiceNumber	No	Merchant Customer Service number displayed on the PayPal checkout.
buyerEmailOptInEnable	No	Enables the buyer to provide their email address on the PayPal pages to be notified of promotions or special events. Possible values are: **0** – Do not enable buyer to provide email. **1** – Enable the buyer to provide email.
noteToBuyer	No	A note from the merchant to the buyer that will be displayed in the PayPal checkout.
paymentAction	No	Defines how to obtain payment. This can be used to override any `captureDelay` setting which can also be used to indicate a **Sale** or **Authorization** only. Possible values are: **Sale** – sale with immediate capture. **Authorization** – authorization subject to later capture. **Order** – order subject to later authorization and capture.
allowedPaymentMethod	No	The payment method type. Specify the value InstantPaymentOnly
insuranceOptionOffered	No	Indicates whether insurance is available as an option the buyer can choose on the PayPal Review page. Possible values are: **true** – The Insurance option displays 'Yes' and the `insuranceAmount`. If true, the

Field Name	Mandatory?	Description
		total shipping insurance for this order must be a positive number. **false** – The Insurance option displays 'No'.
multiShipping	No	Indicates if this payment is associated with multiple shipping addresses. Possible values are: **0** – Single/No shipping address. **1** – Multiple shipping addresses.
noteText	No	Note to the Merchant.
bucketCategoryType	No	The category of a payment. Possible values are: **1** – International shipping **2** – Local delivery **3** – BOPIS, Buy online pick-up in store **4** – PUDO, Pick-up drop-off
locationType	No	Type of merchant location. Required if the items purchased will not be shipped, such as, BOPIS (Buy Online Pick-up In Store) or PUDO (Pick-Up Drop-Off) transactions. Possible values are: **1** – Consumer. **2** – Store, for BOPIS transactions. **3** – PickupDropoff, for PUDO transactions.
locationID	No	Location ID specified by the merchant for BOPIS (Buy Online Pick-up In Store) or PUDO (Pick-Up Drop-Off) transactions.
sellerPayPalAccountID	No	Unique identifier for the Merchant. For parallel payments, this field is required and must contain the Payer Id or the email address of the Merchant.
invNum	No	Merchant's invoice or tracking number.
custom	No	Custom field for your own use.
buyerID	No	The unique identifier provided by eBay for this buyer. The value may or may not be the same as the username. In the case of eBay, it is different.
buyerUsername	No	The user name of the user at the marketplaces site.
buyerRegistrationDate	No	Date when the user registered with the marketplace. In UTC/GMT format; for example, 2013-08-24T05:38:48Z.

Field Name	Mandatory?	Description
allowPushFunding	No	Indicates whether the Merchant can accept push funding. Possible values are: **0** – Merchant cannot accept push funding. **1** – Merchant can accept push funding.
userSelectedFundingSource	No	This element could be used to specify the preferred funding option for a guest user. However, the `landingPage` checkout option must also be set to **Billing**, otherwise, it is ignored. Possible values are: **ChinaUnionPay.** **CreditCard.** **ELV.** **QIWI.**
billingType	No	Type of billing agreement for reference transactions. You must have permission from PayPal to use this field. Possible values are: **MerchantInitiatedBilling** – PayPal creates a billing agreement for each transaction associated with buyer. **MerchantInitiatedBillingSingleAgreement** – PayPal creates a single billing agreement for all transactions associated with buyer. Use this value unless you need per-transaction billing agreements.
billingAgreementDescription	No	Description of goods or services associated with the billing agreement. This field is required for each recurring payment billing agreement. PayPal recommends that the description contain a brief summary of the billing agreement terms and conditions. For example, buyer is billed at "9.99 per month for 2 years".
paymentType	No	Type of PayPal payment you require for the billing agreement. Possible values are: **Any** – The merchant accepts any payment method for the billing agreement, even if it could take a few working days for the movement of funds to the merchant account; this includes echeck, in addition to credit or debit cards and PayPal balance. **InstantOnly** – The payment options accepted by the merchant are credit cards, debit cards or PayPal balance only

Field Name	Mandatory?	Description
		because the merchant expects immediate payment.
taxIDType	No	Buyer's tax ID type. This field is required for Brazil and used for Brazil only. For Brazil use only: The tax ID type is BR_CPF for individuals and BR_CNPJ for businesses.
taxID	No	Buyer's tax ID. This field is required for Brazil and used for Brazil only. For Brazil use only: The tax ID is 11 single-byte characters for individuals and 14 single-byte characters for businesses
returnFMFDetails	No	Flag to indicate whether you want the results returned by Fraud Management Filters when doing a recurring/reference transaction. Possible values are: **0** – Do not receive FMF details (default). **1** – Receive FMF details.
riskSessionCorrelationID	No	The ID of the risk session for correlation purposes when doing a recurring/reference transaction.
merchantSessionID	No	The buyer session identification token when doing a recurring/reference transaction.

[1] PayPal refer to this field as MAXAMT

For further information on the options refer to the PayPal Express Checkout documentation: https://developer.paypal.com/docs/classic/api/merchant/SetExpressCheckout_API_Operation_NVP/.

The option names are case sensitive.

1.4.1 Purchase details

The following request fields may be sent to provided information on the purchased items and to populate the cart on the PayPal checkout.

shippingAmount	No	Shipping costs.
shippingDiscountAmount	No	Discount applied to shipping costs.
handlingAmount	No	Handling costs.
insuranceAmount	No	Insurance costs.
item*XX*Description	No	Description of *XX*th item purchased.
item*XX*Quantity	No	Quantity of *XX*th item purchased.
item*XX*Amount	No	Gross amount for *XX*th item purchased.
item*XX*TaxAmount	No	Tax amount for *XX*th item purchased.
item*XX*ProductCode	No	Product code for *XX*th item purchased.
itemXXProductUrl	No	Shopping cart URL for *XX*th item purchased.
item*XX*Size	No	Size of *XX*th item purchased in the format 'LengthxWidthxHeight Unit'.
item*XX*Weight	No	Weight of *XX*th item purchased in the format 'Weight Unit'.
items	No[1]	Nested array of line items.

[1] Used as an alternative to **itemXXField** format, both formats can not be sent together.

Note: The shopping cart items must total to the amount specified in the transaction or cart items will not be sent to the PayPal checkout.

1.1 Response Fields

1.1.1 Initial Response

These fields will be returned in addition to the request fields from section 5.4.1 and the basic response fields in section 0 minus any card details.

Field Name	Mandatory?	Description
checkoutRef	Yes	Unique reference required to continue this transaction when the PayPal Checkout has completed.
checkoutName	Yes	Unique name of the checkout. For PayPal this is the value **paypal**.
checkoutURL	Yes	URL required to load the PayPal Checkout
checkoutRequest	No	Not required for PayPal.
checkoutOptions	No	Any checkout options passed in the request.
acquirerResponseDetails	Yes	Details about the PayPal response containing any error messages and codes. This can be used along with the normal `responseCode`/`responseMessage` response fields to further determine the reason for any failure.

1.1.1 Continuation Response

These fields will be returned in addition to the request fields from section 14.3.1, the initial response fields in section 14.3.2 and the basic response fields in section 0 minus any card details.

Field Name	Mandatory?	Description
checkoutRef	Yes	Provided if checkoutOnly was used in the continuation response to indicate that a further request will be sent to finalise the transaction.
checkoutName	Yes	Unique name of the checkout. For PayPal this is the value **paypal**.
checkoutDetails	Yes	Associative array or URL encoded array of options used to customise the PayPal Checkout. Refer to section 14.5.3 for values.
customerXXXX	No[1]	Customer details if provided by the PayPal Checkout as documented in section 13.1
deliveryXXX	No[1]	Delivery details if provided by the PayPal Checkout as documented in section 13.4
acquirerResponseDetails	Yes	Details about the PayPal response containing any error messages and codes. This can be used along with the normal `responseCode`/`responseMessage` response fields to further determine the reason for any failure.

[1] The response will include customer/billing address and delivery address details if provided by the PayPal Checkout.

1.1.1 Checkout Details

The following details may be provided in the **`checkoutDetails`** field included in the response. The field will be an array of key/value pairs.

Field Name	Mandatory?	Description
`correlationID`	No	Correlation ID, which uniquely identifies the transaction to PayPal.
`checkoutStatus`	No	Status of the checkout session. If payment is completed, the transaction identification number of the resulting transaction is returned. Possible values are: **PaymentActionNotInitiated** **PaymentActionFailed** **PaymentActionInProgress** **PaymentActionCompleted**
`invNum`	No	Merchant's invoice or tracking number, as set sent in `checkoutDetails.invNum` or assigned by the Gateway.
`custom`	No	Merchant's invoice or tracking number, as set sent in `checkoutDetails.custom` or assigned by the Gateway.
`payPalAdjustment`	No	A discount or gift certificate offered by PayPal to the buyer. This amount is represented by a negative amount. If the buyer has a negative PayPal account balance, PayPal adds the negative balance to the transaction amount, which is represented as a positive value.
`buyerMarketingEmail`	No[1]	Buyer's marketing email address.
`note`	No[2]	Buyer's note to the Merchant.
`cartChangeTolerance`	No	Indicates whether a cart's contents can be modified. If this parameter is not returned, then assume the cart can be modified. This will return **NONE** is financing was used in Germany. Possible values are: **NONE** – The cart cannot be changed. **FLEXIBLE** – The cart can be changed.
`payerID`	No	Buyer's PayPal Customer Account ID.

[1] Only available if email optin was enabled in the initial request using `checkoutOptions.buyerEmailOptInEnable` option.
[2] Only available if the leaving of notes was enabled in the initial request using `checkoutOptions.allowNote` option.

Field Name	Mandatory?	Description
payerStatus	No	Buyer's PayPal status. Possible values are: **verified** **unverified**
billingName	No[1]	Buyer's name. Also returned in `customerName`.
firstName	No[2]	Buyer's first name. Also returned in `customerName`.
middleName	No[5]	Buyer's middle name. Also returned in `customerName`.
lastName	No[5]	Buyer's last name. Also returned in `customerName`.
suffix	No[5]	Buyer's name suffix. Also returned in `customerName`.
business	No	Buyer's business name. Also returned in `customerCompany`.
street	No	Buyer's street first line. Also returned in `customerAddress`.
street2	No	Buyer's street second line. Also returned in `customerAddress`.
city	No	Buyer's city Also returned in `customerTown`.
state	No	Buyer's state. Also returned in `customerCounty`.
zip	No	Buyer's postal code. Also returned in `customerPostcode`.
countryCode	No	Buyer's country code. (ISO 2 char. code) Also returned in `customerCountryCode`.
countryName	No	Buyer's country name.
phoneNum	No	Buyer's contact phone number. Also returned in `customerPhone`.
email	No	Buyer's email address. Also returned in `customerEmail`.
shipToName	No	Name of person/entity to ship to. Also returned in `deliveryName`.

[1] Permission is needed from PayPal to support this field.
[2] These fields are used when no permission to use `billingName`.

Field Name	Mandatory?	Description
shipToStreet	No	Ship to street first line. Also returned in `deliveryAddress`.
shipToStreet2	No	Ship to street second line. Also returned in `deliveryAddress`.
shipToCity	No	Ship to city. Also returned in `deliveryTown`.
shipToState	No	Ship to state. Also returned in `deliveryCounty`.
shipToZip	No	Ship to postal code. Also returned in `deliveryPostcode`.
shipToCountryCode	No	Ship to country code. (ISO 2 char. code) Also returned in `deliveryCountryCode`.
shipToCountryName	No	Ship to country name.
shipToPhoneNum	No	Ship to phone number. Also returned in `deliveryPhone`.
shipToAddressStatus	No	Status of shipping address on file with PayPal. Possible values are: **none** **Confirmed** **Unconfirmed**
addressNormalizationStatus	No	The PayPal address normalization status for Brazilian addresses. Possible values are: **None** **Normalized** **Unnormalized** **UserPreferred**
amount	No	Total amount for this order.
itemAmount	No	Total item amount for this order.
taxAmount	No	Tax amount for this order.
exchangeRate	No	Exchange rate for this order.
shippingAmount	No	Shipping amount for this order.
handlingAmount	No	Handling amount for this order.
insuranceAmount	No	Insurance amount for this order.
shipDiscountAmount	No	Shipping discount amount for this order.

Field Name	

Field Name	Mandatory?	Description
`instrumentCategory`	No	Identifies the category of the promotional payment instrument. Possible values are: **1** – PayPal Credit® (formerly Bill Me Later®). **2** – A Private Label Credit Card (PLCC) or co-branded payment card.
`instrumentID`	No	An instrument ID (issued by the external party) corresponding to the funding source used in the payment.
`shippingCalculationMode`	No	Describes how the options that were presented to the buyer were determined. Possible values are: **API – Callback** **API – Flatrate**
`insuranceOptionSelected`	No	The option that the buyer chose for insurance. Possible values are: **Yes –** opted for insurance. **No –** did not opt for insurance.
`shippingOptionIsDefault`	No	Indicates whether the buyer chose the default shipping option. Possible values are: **true –** chose the default shipping option. **false –** did not choose the default shipping option.
`shippingOptionAmount`	No	The shipping amount that the buyer chose.
`shippingOptionName`	No	The name of the shipping option, such as Air or Ground.
`scheduledShippingDate`	No	The scheduled shipping date is returned only if scheduled shipping options are passed in the request.
`scheduledShippingPeriod`	No	The scheduled shipping period is returned only if scheduled shipping options are passed in the request.
`sellerPayPalAcountID`	No	Unique identifier for the merchant. For parallel payments, this field contains either the Payer Id or the email address of the merchant.
`taxIDType`	No	Buyer's tax ID type. This field is required for Brazil and used for Brazil only.

Field Name	Mandatory?	Description
		For Brazil use only: The tax ID type is BR_CPF for individuals and BR_CNPJ for businesses.
taxID	No	Buyer's tax ID. This field is required for Brazil and used for Brazil only. For Brazil use only: The tax ID is 11 single-byte characters for individuals and 14 single-byte characters for businesses
billingAgreementID	No	Identification number of the billing agreement. When the buyer approves the billing agreement, it becomes valid and remains valid until it is cancelled by the buyer.
billingAgreementAcceptedStatus	No	Indicates whether the buyer accepted the billing agreement for a recurring payment. Currently, this field is always returned in the response for agreement based products, such as, subscriptions, reference transactions and recurring payments, as well as for regular single payment transactions. **0 –** Not accepted. **1 –** Accepted.
paymentStatus	No	Status of the payment. Possible values are: **None –** No status. **Canceled-Reversal –** A reversal has been cancelled, for example, when you win a dispute and the funds for the reversal have been returned to you. **Completed –** The payment has been completed, and the funds have been added successfully to your account balance. **Denied –** You denied the payment. This happens only if the payment was previously pending because of possible reasons described for the `pendingReason` element. **Expired –** The authorization period for this payment has been reached. **Failed –** The payment has failed. This happens only if the payment was made from your buyer's bank account. **In-Progress –** The transaction has not terminated; for example, an authorization may be awaiting completion. **Partially-Refunded –** The payment has been partially refunded.

Field Name	Mandatory?	Description
		Pending – The payment is pending. See the `pendingReason` field for more information. **Refunded –** You refunded the payment. **Reversed –** A payment was reversed due to a charge back or other type of reversal. The funds have been removed from your account balance and returned to the buyer. The reason for the reversal is specified in the `reasonCode` element. **Processed –** A payment has been accepted. **Voided –** An authorization for this transaction has been voided.
`refundStatus`	No	Status of the refund. Possible value are: none **–** returned if the refund fails instant – refund was instant delayed – refund was delayed
`pendingReason`	No[1]	The reason the payment is pending. Possible values are: **none –** No pending reason. **address –** The payment is pending because your buyer did not include a confirmed shipping address and your Payment Receiving Preferences is set such that you want to manually accept or deny each of these payments. To change your preference, go to the Preferences section of your Profile. **authorization –** The payment is pending because it has been authorized but not settled. You must capture the funds first. **echeck –** The payment is pending because it was made by an eCheck that has not yet cleared. **intl –** The payment is pending because you hold a non-U.S. account and do not have a withdrawal mechanism. You must manually accept or deny this payment from your Account Overview. **multi-currency –** You do not have a balance in the currency sent, and you do not have your Payment Receiving Preferences set to automatically convert and accept this payment. You must manually accept or deny this payment. **order –** The payment is pending because it is part of an order that has been authorized but not settled.

[1] `pendingReason` is returned in the response only if **`paymentStatus`** is **Pending**.

Field Name	Mandatory?	Description
		payment-review – The payment is pending while it is being reviewed by PayPal for risk. **regulatory-review –** The payment is pending while we make sure it meets regulatory requirements. You will be contacted again in from 24 to 72 hours with the outcome of the review. **unilateral –** The payment is pending because it was made to an email address that is not yet registered or confirmed. **verify –** The payment is pending because you are not yet verified. You must verify your account before you can accept this payment. **other –** The payment is pending for a reason other than those listed above. For more information, contact PayPal Customer Service.
`reasonCode`	No	The reason for a reversal if the transaction type is reversal. Possible values are: **none –** No reason code. **chargeback –** A reversal has occurred on this transaction due to a charge back by your buyer. **guarantee –** A reversal has occurred on this transaction due to your buyer triggering a money-back guarantee. **buyer-complaint –** A reversal has occurred on this transaction due to a complaint about the transaction from your buyer. **refund –** A reversal has occurred on this transaction because you have given the buyer a refund. **other –** A reversal has occurred on this transaction due to a reason not listed above.
`protectionEligibilityType`	No	The kind of seller protection in force for the transaction. Possible value are: **ItemNotReceivedEligible –** Merchant is protected by PayPal's Seller Protection Policy for Item Not Received. **UnauthorizedPaymentEligible –** Merchant is protected by PayPal's Seller Protection Policy for Unauthorized Payment. **Ineligible –** Merchant is not protected under the Seller Protection Policy. (Multiple values are separated by commas)

Field Name	Mandatory?	Description
feeAmount	No	PayPal fee amount charged for the transaction.
settleAmount	No	Amount deposited in your PayPal account after a currency conversion.
storeID	No	StoreId as entered in the transaction.
terminalID	No	TerminalId as entered in the transaction.

1.2 Transaction Lifecycle

PayPal transactions will use the normal Authorise, Capture life cycle as document in appendix A-12.1 with the following differences. In addition, the PayPal **`paymentAction`** option can be included in the **`checkoutOptions`** field to further alter the normal payment lifecycle to allow an Order, Authorise, Capture model to be specified or a straight Sale model.

1.2.1 Order

If a **`paymentAction`** with a value of 'Order' is sent, then PayPal will store the transaction but delay authorising it until instructed. To instruct PayPal to authorise the transaction a further management request can be sent to the Gateway with an **`action`** of 'AUTHORISE' and the **`xref`** of the transaction to authorise, alternatively the AUTHORISE command can be selected in the Merchant Management System (MMS). The transaction will be left in the 'received' state.

1.2.2 Authorise

If no **`paymentAction`** is specified or a **`paymentAction`** with a value of 'Authorize' is sent, then PayPal will authorise the transaction on receipt as per a standard card transaction and the Merchant can capture it later if they used the **`captureDelay`** field.

For the first three days (by default) of the authorisation, funds are reserved. This is known as the honour period. After the honour period, captures can still be attempted, but may be returned with insufficient funds.

Authorisations have a fixed expiry period of 29 days.

1.2.3 Sale

If a **`paymentAction`** with a value of 'Sale' is sent then PayPal will immediately capture the transaction after authorisation. The transaction will be regarded as having been settled and the Merchant will not be able to capture it manually and it will not be sent for settlement that evening. The transaction will be left in either the **accepted** or **rejected** terminal states depending on whether PayPal accepted or rejected the transaction.

1.2.4 Capture

Transactions which have been authorised by PayPal and not immediately settled due to a **`paymentAction`** of 'Sale' will be able to be captured as normal by the Merchant.

Captures are sent to PayPal immediately and the PayPal response and the transaction will be left in either the **accepted** or **rejected** terminal state depending on whether PayPal accepted or rejected the capture request.

There is no need to wait for the nightly settlement batch to run as with normal card transactions. This means that it is not possible to change the amount to be captured or cancel the transaction one a capture has been requested.

Note: PayPal allows multiple captures in a different manner to the Gateway where they sum the individual capture amounts. This mode of operation is not possible using the Gateway and only a single capture operation can be processed.

1.2.5 Refund

PayPal transactions can be refunded the same as normal card transactions however, like capture requests, these will be sent to PayPal immediately and not batched up and sent as part of the nightly settlement process. This means the transaction will be left in either the **accepted** or **rejected** terminal state depending on whether PayPal accepted or rejected the refund request.

Refunds can be made for full or partial amounts, with multiple refunds allowed up to the original authorised amount.

By default, PayPal allows a Merchant up to 60 days from the original authorised transaction date to perform refunds.

1.2.6 Cancel

The Merchant should cancel any transactions they do not wish to capture so as to prevent 'pending' transactions on the Customers PayPal account.

Authorisations should be cancelled when an initial authorisation was created to confirmed the validity of funds during checkout, but the goods will not ship for a significant amount of time (>29 days). Cancelling the transaction will mean that the Merchant will have to contact the Customer for an alternative payment method.

All transactions must be completed by being captured or cancelled.

1.2.7 Pending Payments

PayPal may put a transaction into a pending state when flagged for additional fraud review. This state is known to PayPal as payment review or IPR.

IPR transaction will be automatically cancelled by the Gateway and treated as referred transactions with a **`responseCode`** of **2** and a **`responseMessage`** indicating the reason the transaction was put into a pending state. Unlike card referred transactions an authorisation code cannot be obtained from PayPal verbally and the transaction resent.

1.3 Reference Transactions

PayPal does not allow ad hoc 'card on file' type repeat or recurring transactions using the **`xref`** of a reference transaction unless that transaction has specifically started a PayPal Billing Agreement.

If the Merchant wants to be able to make future repeat or recurring transactions, then the initial transaction must include the **`billingType`** and **`billingAgreementDescription`** options in the **`checkoutOptions`** so as to identify this transaction as the start of a recurring billing sequence.

This will cause the Gateway to request PayPal setup a Billing Agreement between the Merchant and the Customer. In this case the PayPal Billing Agreement ID will be returned as part of the **`checkoutDetails`** and displayed on the Merchant Management System (MMS) as part of the payment details so that the Merchant can easily see which PayPal transactions can be used for recurring billing.

A-1 Response Codes

The Gateway will always issue a **`responseCode`** to report the status of the transaction. These codes should be used rather than the **`responseMessage`** field to determine the outcome of a transaction.

A zero response code always indicates a successful outcome.

Response codes are grouped as follows, the groupings are for informational purposes only and not all codes in a group are used;

Acquirer (FI) Error codes: 1-99	
Code	**Description**
0	Successful / authorised transaction. Any code other than **0** indicates an unsuccessful transaction
2	Card referred
4	Card declined – keep card
5	Card declined
30	An error occurred. Check **`responseMessage`** for more detail

General Error Codes: 65536 - 65791	
Code	**Description**
65536	Transaction in progress. Contact customer support if this error occurs
65537	Reserved for future use. Contact customer support if this error occurs
65538	Reserved for future use. Contact customer support if this error occurs
65539	Invalid Credentials: **`merchantID`** is unknown
65540	Permission denied: caused by sending a request from an unauthorised IP address
65541	Action not allowed: the transaction state or Acquirer doesn't support this action
65542	Request Mismatch: fields sent while completing a request do not match initially requested values. Usually due to sending different card details to those used to authorise the transaction when completing a 3-D Secure transaction or performing a REFUND_SALE transaction.
65543	Request Ambiguous: request could be misinterpreted due to inclusion of mutually exclusive fields
65544	Request Malformed: couldn't parse the request data

General Error Codes: 65536 - 65791	
Code	**Description**
65545	Suspended Merchant account
65546	Currency not supported by Merchant
65547	Request Ambiguous, both **`taxValue`** and **`discountValue`** provided when should be one only
65548	Database error
65549	Payment processor communications error
65550	Payment processor error
65551	Internal Gateway communications error
65552	Internal Gateway error
65553	Encryption error.
65554	Duplicate request. Refer to Section 1.
65555	Settlement error.
65556	AVS/CV2 Checks are not supported for this card (or Acquirer)
65557	IP Blocked: Request is from a banned IP address
65558	Primary IP blocked: Request is not from one of the primary IP addresses configured for this Merchant Account
65559	Secondary IP blocked: Request is not from one of the secondary IP addresses configured for this Merchant Account
65560	Reserved for future use. Contact customer support if this error occurs
65561	Unsupported Card Type: Request is for a card type that is not supported on this Merchant Account
65562	Unsupported Authorisation: External authorisation code **`authCode`** has been supplied and this is not supported for the transaction or by the Acquirer
65563	Request not supported: The Gateway or Acquirer does not support the request
65564	Request expired: The request can not be completed as the information is too old
65565	Request retry: The request can be retried later
65566	Test Card Used: A test card was used on a live Merchant Account
65567	Unsupported card issuing country: Request is for a card issuing country that is not supported on this Merchant Account
65568	Unsupported payment type: Request uses a payment type which is not supported on this Merchant Account

3-D Secure Error Codes: 65792 - 66047	
Code	**Description**
65792	3-D Secure transaction in progress. Contact customer support if this error occurs
65793	Unknown 3-D Secure Error
65794	3-D Secure processing is unavailable. Merchant account doesn't support 3-D Secure
65795	3-D Secure processing is not required for the given card
65796	3-D Secure processing is required for the given card
65797	Error occurred during 3-D Secure enrolment check
65798	Reserved for future use. Contact customer support if this error occurs
65799	Reserved for future use. Contact customer support if this error occurs
65800	Error occurred during 3-D Secure authentication check
65801	Reserved for future use. Contact customer support if this error occurs
65802	3-D Secure authentication is required for this card
65803	3-D Secure enrolment or authentication failure and Merchant 3-D Secure preferences are to STOP processing

Missing Request Field Error Codes: 66048 - 66303	
Code	**Description**
66048	Missing request. No data posted to integration URL
66049	Missing **`merchantID`** field
66050	Reserved for future use. Contact customer support if this error occurs
66051	Reserved for internal use. Contact customer support if this error occurs
66052	Reserved for internal use. Contact customer support if this error occurs
66053	Reserved for internal use. Contact customer support if this error occurs
66054	Reserved for internal use. Contact customer support if this error occurs
66055	Missing **`action`** field
66056	Missing **`amount`** field
66057	Missing **`currencyCode`** field
66058	Missing **`cardNumber`** field

Missing Request Field Error Codes: 66048 - 66303	
Code	**Description**
66059	Missing **cardExpiryMonth** field
66060	Missing **cardExpiryYear** field
66061	Missing **cardStartMonth** field (reserved for future use)
66062	Missing **cardStartYear** field (reserved for future use)
66063	Missing **cardIssueNumber** field (reserved for future use)
66064	Missing **cardCVV** field
66065	Missing **customerName** field
66066	Missing **customerAddress** field
66067	Missing **customerPostCode** field
66068	Missing **customerEmail** field
66069	Missing **customerPhone** field (reserved for future use)
66070	Missing **countyCode** field
66071	Missing **transactionUnique** field (reserved for future use)
66072	Missing **orderRef** field (reserved for future use)
66073	Missing **remoteAddress** field (reserved for future use)
66074	Missing **redirectURL** field
66075	Missing **callbackURL** field (reserved for future use)
66076	Missing **merchantData** field (reserved for future use)
66077	Missing **origin** field (reserved for future use)
66078	Missing **duplicateDelay** field (reserved for future use)
66079	Missing **itemQuantity** field (reserved for future use)
66080	Missing **itemDescription** field (reserved for future use)
66081	Missing **itemGrossValue** field (reserved for future use)
66082	Missing **taxValue** field (reserved for future use)
66083	Missing **discountValue** field (reserved for future use)
66084	Missing **taxDiscountDescription** field (reserved for future use)

Missing Request Field Error Codes: 66048 - 66303	
Code	**Description**
66085	Missing **xref** field (reserved for future use)
66086	Missing **type** field (reserved for future use)
66087	Missing **signature** field (field is required if message signing is enabled)
66088	Missing **authorisationCode** field (reserved for future use)
66089	Missing **transactionID** field (reserved for future use)
66090	Missing **threeDSRequired** field (reserved for future use)
66091	Missing **threeDSMD** field (reserved for future use)
66092	Missing **threeDSPaRes** field
66093	Missing **threeDSECI** field
66094	Missing **threeDSCAVV** field
66095	Missing **threeDSXID** field
66096	Missing **threeDSEnrolled** field
66097	Missing **threeDSAuthenticated** field
66098	Missing **threeDSCheckPref** field
66099	Missing **cv2CheckPref** field
66100	Missing **addressCheckPref** field
66101	Missing **postcodeCheckPref** field
66102	Missing **captureDelay** field
66103	Missing **orderDate** field
66104	Missing **grossAmount** field
66105	Missing **netAmount** field
66016	Missing **taxRate** field
66016	Missing **taxReason** field
66160	Missing **cardExpiryDate** field
66161	Missing **cardStartDate** field

Invalid Request Field Error Codes: 66304 - 66559	
Code	**Description**
66304	Invalid request
66305	Invalid **merchantID** field
66306	Reserved for future use. Contact customer support if this error occurs
66307	Reserved for internal use. Contact customer support if this error occurs
66308	Reserved for internal use. Contact customer support if this error occurs
66309	Reserved for internal use. Contact customer support if this error occurs
66310	Reserved for internal use. Contact customer support if this error occurs
66311	Invalid **action** field
66312	Invalid **amount** field
66313	Invalid **currencyCode** field
66314	Invalid **cardNumber** field
66315	Invalid **cardExpiryMonth** field
66316	Invalid **cardExpiryYear** field
66317	Invalid **cardStartMonth** field
66318	Invalid **cardStartYear** field
66319	Invalid **cardIssueNumber** field
66320	Invalid **cardCVV** field
66321	Invalid **customerName** field
66322	Invalid **customerAddress** field
66323	Invalid **customerPostCode** field
66324	Invalid **customerEmail** field
66325	Invalid **customerPhone** field
66326	Invalid **countyCode** field
66327	Invalid **transactionUnique** field (reserved for future use)
66328	Invalid **orderRef** field (reserved for future use)
66329	Invalid **remoteAddress** field
66330	Invalid **redirectURL** field

Invalid Request Field Error Codes: 66304 - 66559	
Code	**Description**
66331	Invalid **callbackURL** field (reserved for future use)
66332	Invalid **merchantData** field (reserved for future use)
66333	Invalid **origin** field (reserved for future use)
66334	Invalid **duplicateDelay** field. Refer to Section 1.
66335	Invalid **itemQuantity** field
66336	Invalid **itemDescription** field
66337	Invalid **itemGrossValue** field
66338	Invalid **taxValue** field
66339	Invalid **discountValue** field
66340	Invalid **taxDiscountDescription** field (reserved for future use)
66341	Invalid **xref** field
66342	Invalid **type** field
66343	Invalid **signature** field
66344	Invalid **authorisationCode** field
66345	Invalid **transactionID** field
66356	Invalid **threeDSRequired** field
66347	Invalid **threeDSMD** field
66348	Invalid **threeDSPaRes** field
66349	Invalid **threeDSECI** field
66350	Invalid **threeDSCAVV** field
66351	Invalid **threeDSXID** field
66352	Invalid **threeDSEnrolled** field
66353	Invalid **threeDSAuthenticated** field
66354	Invalid **threeDSCheckPref** field
66355	Invalid **cv2CheckPref** field
66356	Invalid **addressCheckPref** field

Invalid Request Field Error Codes: 66304 - 66559	
Code	**Description**
66357	Invalid **`postcodeCheckPref`** field
66358	Invalid **`captureDelay`** field.
66359	Invalid **`orderDate`** field
66360	Invalid **`grossAmount`** field
66361	Invalid **`netAmount`** field
66362	Invalid **`taxRate`** field
66363	Invalid **`taxReason`** field
66416	Invalid card expiry date. Must be a date sometime in the next 10 years
66417	Invalid card start date. Must be a date sometime in the last 10 years

A-2 AVS / CV2 Check Response Codes

The AVS/CV2 Check Response Message field **`avscv2ResponseMessage`** is sent back in the raw form that is received from the Acquiring bank and can contain the following values;

Response	Description
ALL MATCH	AVS and CV2 match
SECURITY CODE MATCH ONLY	CV2 match only
ADDRESS MATCH ONLY	AVS match only
NO DATA MATCHES	No matches for AVS and CV2
DATA NOT CHECKED	Supplied data not checked
SECURITY CHECKS NOT SUPPORTED	Card scheme does not support checks

The AVS/CV2 Response Code **`avscv2ResponseCode`** is made up of six characters and is sent back in the raw form that is received from the Acquiring bank. The first 4 characters can be decoded as below, the remaining 2 characters are currently reserved for future use;

Position 1 Value	Description
0	No Additional information available.
1	CV2 not checked
2	CV2 matched.
4	CV2 not matched
8	Reserved

Position 2 Value	Description
0	No Additional information available.
1	Postcode not checked
2	Postcode matched.
4	Postcode not matched
8	Postcode partially matched

Position 3 Value	Description
0	No Additional Information
1	Address numeric not checked
2	Address numeric matched
4	Address numeric not matched
8	Address numeric partially matched

Position 4 Value	Description
0	Authorising entity not known
1	Authorising entity – merchant host
2	Authorising entity – acquirer host
4	Authorising entity – card scheme
8	Authorising entity – issuer

A-3 3-D Secure Enrolment/Authentication Codes

The 3-D Secure enrolment check field **`threeDSEnrolled`** can return the following values;

Y - Enrolled: The card is enrolled in the 3-D Secure program and the payer is eligible for authentication processing.

N - Not Enrolled: The checked card is eligible for the 3-D Secure (it is within the card association's range of accepted cards) but the card issuing bank does not participate in the 3-D Secure program. If the Cardholder later disputes the purchase, the issuer may not submit a chargeback to the Merchant.

U - Unable To Verify Enrolment: The card associations were unable to verify if the Cardholder is registered. As the card is ineligible for 3-D Secure, Merchants can choose to accept the card nonetheless and precede the purchase as non-authenticated and submits authorization with ECI 7. The Acquirer/Merchant retains liability if the Cardholder later disputes making the purchase.

E - Error Verify Enrolment: The Gateway encountered an error. This card is flagged as 3-D Secure ineligible. The card can be accepted for payment, yet the Merchant may not claim a liability shift on this transaction in case of a dispute with the Cardholder.

The 3-D Secure authentication check field **`threeDSAuthenticated`** can return the following values;

Y - Authentication Successful: The Issuer has authenticated the Cardholder by verifying the identity information or password. A CAVV and an ECI of 5 is returned. The card is accepted for payment.

N - Not Authenticated: The Cardholder did not complete authentication and the card should not be accepted for payment.

U - Unable To Authenticate: The authentication was not completed due to technical or another problem. A transmission error prevented authentication from completing. The card should be accepted for payment but no authentication data will be passed on to authorization processing and no liability shift will occur.

A - Attempted Authentication: A proof of authentication attempt was generated. The Cardholder is not participating, but the attempt to authenticate was recorded. The card should be accepted for payment and authentication information passed to authorization processing.

E - Error Checking Authentication: The Gateway encountered an error. The card should be accepted for payment but no authentication information will be passed to authorization processing and no liability shift will occur.

A-4 3-D Secure Enrolment/Authentication Only

Normally the Gateway will perform most of the 3-D Secure processing in the background leaving the only the actual contacting of the issuers Access Control Server (ACS) to the Merchant.

However, there may be times when you may wish to gain more control over the Enrolment and Authentication process. The following field allows the request processing to stop after the 3-D Secure enrolment check or authentication check and return;

Field Name	Mandatory?	Description
threeDSOnly	No	Complete the processing as far as the next 3-D Secure stage and then return with the appropriate response fields for that stage.

As this stop is requested then a **responseCode** is returned as **0 (Success)** however it will be recorded in the Merchant Management System (MMS) as **65792 (3DS IN PROGRESS)** indicating that the transaction has been prematurely halted expecting it to be continued to the next 3-D Secure stage when required. In order to continue the process, the **threeDSMD** field is returned along with any relevant 3-D Secure response fields suitable for that stage in the processing.

If this flag is used when 3-D Secure is not enabled on the account or after the 3-D Secure process has been completed for the request (i.e. once the authentication step has completed), then passing the flag will cause the transaction to abort with a **responseCode** of **65795 (3DS PROCESSING NOT REQUIRED)**. This ensures that the transaction doesn't go on to completion by accident while trying do 3-D Secure enrolment or authentication only.

A-5 Request Checking Only

Sometimes you may wish to submit a request to the Gateway in order for it to be validated only and not processed or sent to the Acquirer. In these instances, the following flag can be used which will stop the processing after the integrity verification has been performed;

Field Name	Mandatory?	Description
`checkOnly`	No	Check the request for syntax and field value errors only. Do not attempt to submit the transaction for honouring by the Merchants financial institution.

If the request is ok, then a `responseCode` is returned as **0 (Success)** otherwise the code that would have prevented the request from completing is returned.

Note: *in these situations, the request is not stored by the Gateway and is not available within the Merchants Management System (MMS).*

A-6 Merchant Account Mapping

Merchant Accounts can be grouped together so that if a transaction is sent to an account that doesn't support either the requested card type or currency then it can be automatically routed to another account in the same group that does support them.

For example; you can group a Merchant Account that only supports American Express cards with a Merchant Account that only supports Visa cards, then if a request using an American Express card is sent to the Visa only Merchant Account the Gateway will automatically route it to the American Express Merchant Account.

This prevents you from needing to know the card type in advance in order to send the request to the correct Merchant Account. This is important when using the Hosted integration as you don't know the card type at the time you send the request.

It is usual for you to have one master account to which you direct all requests and then group all your accounts together.

Any Gateway routing of the transaction can be seen from the following additional response fields;

Field Name	Returned?	Description
requestMerchantID	Always	ID of Merchant Account request was sent to (usually same as **merchantID**).
processMerchantID	Always	ID of Merchant Account request was processed by.

A-7 Velocity Control System (VCS)

The Gateway allows you to configure velocity controls using the Merchant Management System (MMS). These can be used to automatically email you decline transactions that exceed these controls.

For example; you can set up a control that stops a certain card number from being used more than twice in the space of a few minutes.

If one or more of these controls are broken by a transaction, then the following response fields will show the problem.

If a transaction is declined due to one or more of these rules, then a **`responseCode`** of **5 (VCS DECLINE)** will be returned.

Field Name	Returned?	Description
`vcsResponseCode`	Always	VCS error code. Normally **5**. Refer to appendix A-1 for details.
`vcsResponseMessage`	Always	Description of above response code or list of controls that broken by this transaction.

A-8 Capture Delay

Capture Delay enables the Merchant to specify a delay between the authorisation of a payment and its capture. This allows you time to verify the order and choose whether to fulfil it or cancel it. This can be very helpful in preventing chargebacks due to fraud.

When NOT using capture delay, payments are authorised and captured immediately - funds are automatically debited from the Customer's credit or debit card at that time.

When using capture delay, the payment is authorised only at the time of payment - funds are reserved against the credit or debit card and will not be debited until the payment is captured or cancelled.

The Customer experience with capture delay is exactly the same as when capture delay is not used. The Customer will not know whether you are using capture delay or not.

If you choose to use capture delay, you specify the number of days that capture is delayed for - this will be in the range of 0 - 30 days. Payments will automatically be captured after that delay unless you manually cancel the transaction (either using the Hosted Integration or via the Merchant Management System (MMS)). (Note that some cards require capture within 4-5 days - if payment is not automatically captured within that 4-5 day period, the transaction will expire and the reserved funds will be released to the Customer.)

Why Use Capture Delay?

Capture delay allows you to accept online orders normally, but allows you to cancel any transactions that you cannot or will not fulfil, thereby reducing the risks of chargeback. If you receive an order that appears to be fraudulent or that you cannot or do not wish to fulfil, you can simply cancel the transaction.

Note: Cancelling a transaction will not reverse the authorisation and will not release the funds back to the Customer. The authorisation will be left to expire and release reserved funds, the time taken for this varies between cards.

Some Acquirers do not support delayed capture, in which the Hosted Integration will return a ***`responseCode`*** *of* ***66358 (INVALID CAPTURE DELAY)****.*

A-9 Types of card

The following is a list of primary card types supported by the Gateway.

Card Code	Card Type
MC	MasterCard Credit
MD	MasterCard Debit
MA	MasterCard International Maestro
MI	MasterCard/Diners Club
MP	MasterCard Purchasing
MU	MasterCard Domestic Maestro (UK)
VC	Visa Credit
VD	Visa Debt
EL	Visa Electron
VA	Visa ATM
VP	Visa Purchasing
AM	American Express
JC	JCB

The Gateway primarily supports MasterCard, Visa and American Express branded cards. Some Acquirers may support JCB cards. Not all Acquirers support all types.

Where cards are provided by a single card scheme then the primary card code is also used as a code to identify the card scheme (referred to as the **`cardSchemeCode`** in the transaction response). For example, cards issued by VISA will use the code '**VC**', cards issued by MasterCard will use the code '**MC**', etc.

The following is a list of secondary card types recognised by the Gateway.

Card Code	Card Type
CF	Clydesdale Financial Services
CU	China UnionPay
BC	BankCard
DK	Dankort
DS	Discover
DI	Diners Club
DE	Diners Club Enroute
DC	Diners Club Carte Blanche
FC	FlexCache
LS	Laser
SO	Solo
ST	Style
SW	Switch
TP	Tempo Payments
IP	InstaPayment
XX	Unknown/unrecognised card type

These cards may be returned in response to a card lookup but they are either deprecated or most likely not supported by any current Acquirer.

A-10 Integration Testing

You can perform test transaction using one of the test Merchant IDs below and using test card details.

For non 3-D Secure testing use Merchant ID **101093**
For 3-D Secure Testing use Merchant ID **101094**

Test Merchant Accounts are not connected to an Acquirer and so simulate their response depending on the request **`amount`** as follows;

Amount range from	Amount range to	Expected response
101 (£1.01)	4999 (£49.99)	AUTH CODE: XXXXXX
5000 (£50.00)	9999 (£99.99)	CARD REFERRED
10000 (£100.00)	14999 (£149.99)	CARD DECLINED
15000+ (£150.00+)		CARD DECLINED – KEEP CARD

A-10.1 Test Card Details

DO NOT USE THESE TEST CARDS ON LIVE MERCHANT ACCOUNT. THEY ARE FOR TEST PURPOSES ONLY.

The expiry date used for each test card should be December of the current year; in two digit format – E.g. 12/15 for December 2015

Visa Credit

Card Number	CVV	Address
4929421234600821	356	Flat 6 Primrose Rise 347 Lavender Road Northampton NN17 8YG
4543059999999982	110	76 Roseby Avenue Manchester M63X 7TH
4543059999999990	689	23 Rogerham Mansions 4578 Ermine Street Borehamwood WD54 8TH

Visa Debit

Card Number	CVV	Address
4539791001730106	289	Unit 5 Pickwick Walk 120 Uxbridge Road Hatch End Middlesex HA6 7HJ
4462000000000003	672	Mews 57 Ladybird Drive Denmark 65890

MasterCard Credit

Card Number	CVV	Address
5301250070000191	419	25 The Larches Narborough Leicester LE10 2RT
5413339000001000	304	Pear Tree Cottage The Green Milton Keynes MK11 7UY
5434849999999951	470	34a Rubbery Close Cloisters Run Rugby CV21 8JT
5434849999999993	557	4-7 The Hay Market Grantham NG32 4HG

MasterCard Debit

Card Number	CVV	Address
5573 4712 3456 7898	159	Merevale Avenue Leicester LE10 2BU

UK Maestro

Card Number	CVV	Address
6759 0150 5012 3445 002	309	The Parkway 5258 Larches Approach Hull North Humberside HU10 5OP
6759 0168 0000 0120 097	701	The Manor Wolvey Road Middlesex TW7 9FF

JCB

Card Number	CVV	Address
3540599999991047	209	2 Middle Wallop Merideth-in-the-Wolds Lincolnshire LN2 8HG

Electron

Card Number	CVV	Address
4917480000000008	009	5-6 Ross Avenue Birmingham B67 8UJ

American Express

Card Number	CVV	Address
374245455400001	4887	The Hunts Way Southampton SO18 1GW

Diners Club

Card Number	CVV Number
36432685260294	**111**

A-10.2 Test 3-D Secure Card Details

DO NOT USE THESE TEST CARDS ON LIVE MERCHANT ACCOUNT.
THEY ARE FOR TEST PURPOSES ONLY.

The expiry date used for each test card should be December of the current year; in two digit format – E.g. 12/15 for December 2015

Visa Test Cards

Card Number	CVV	Address	Postcode	Amount	Test Scenario
4909630000000008				£12.01	Card range not participating
4012010000000000009				£12.02	Card registered with VbV (automated ACS response – click on Submit button)
4012001037141112	083	16	155	£12.03	Card registered with Visa (automated ACS response – click on Submit button)
4012001037484447	450	200	19	£12.04	Failed authentication – issuer database unavailable
4015501150000216				£12.05	Attempts processing (automated ACS response – click on Submit button)

MasterCard Test Cards

Note: These test cards are controlled by MasterCard and won't always act as expected. The 3-D Secure passwords can be changed by anyone during the 3-D Secure testing which means the password won't then work for the next person. The standard fall-back password is dog33cat. Use Visa's 3-D Secure test cards if these are not behaving as expected.

Card Number	CVV	Address	Postcode	Amount	Test Scenario
503396198900000818	332	31	18	£11.01	Enrolled International Maestro account number – valid SecureCode (multiple cardholder). Select 'MEGAN SANDERS' with SecureCode password: secmegan1
5453010000070789	508	20	52	£11.02	Enrolled account number - valid SecureCode (single) SecureCode password: sechal1
5453010000070151	972	22	08	£11.03	Enrolled account number – mixed SecureCode (multi) SecureCode password: Hannah – sechannah1 (bad) Haley – sechaley1 (good)
5453010000070284	305	35	232	£11.04	Enrolled account number – invalid SecureCode Invalid SecureCode password: invseccode
5453010000084103	470	73	170	£11.05	Attempts processing
5453010000070888	233	1	248	£11.06	Account number not enrolled
5199992312641465	006	21	14	£11.07	Card range not participating

A-10.3 PayPal Sandbox Accounts

PayPal testing is available on the standard **101093** test Merchant account however Merchant's may wish to contact their support representative to have their own PayPal test Merchant account created which connects to the Merchant's own PayPal sandbox account enabling them to view the transactions as they are sent to PayPal.

A-11 Sample Signature Calculation

It is highly recommended that transactions are protected using message signing. The signing process offers a quick and simple way to ensure that the message came from an authorised source and has not been tampered with during transmission.

Signing however must be done on your servers and never left for the Customers browser to do in JavaScript as this would mean revealing your secret signature code to anyone who viewed the JavaScript code in the browser.

Signatures are especially important when a transaction is sent from a browsers payment form via the use of hidden for fields as the Customer can easily use tools built into their browser to modify these hidden fields and maybe change things like the amount they should be charged etc.

The section below gives a step by step example of how to sign a transaction complete with coding examples using the PHP language.

Example Signature Key:

```
$key = 'DontTellAnyone'
```

Example Transaction:

```
$tran = array (
  'merchantID' => '101093',
  'action' => 'SALE',
  'type' => '1',
  'currencyCode' => '826',
  'countryCode' => '826',
  'amount' => '2691',
  'transactionUnique' => '55f025addd3c2',
  'orderRef' => 'Signature Test',
  'cardNumber' => '4929 4212 3460 0821',
  'cardExpiryDate' => '1213',
)
```

The transaction used for signature calculation must not include any 'signature' field as this will be added after signing once its value is known.

Step 1 - Sort transaction values by their field name

Transaction fields must be in ascending field name order according to their numeric ASCII value.

```
ksort($tran);
```

```
array ( 'action' => 'SALE', 'amount' => '2691', 'cardExpiryDate' =>
'1213', 'cardNumber' => '4929 4212 3460 0821', 'countryCode' =>
'826', 'currencyCode' => '826', 'merchantID' => '101093', 'orderRef'
=> 'Signature Test', 'transactionUnique' => '55f025addd3c2', 'type'
=> '1' )
```

Step 2 - Create url encoded string from sorted fields

Use RFC 1738 and the application/x-www-form-urlencoded media type, which implies that spaces are encoded as plus (+) signs.

```
$str = http_build_query($tran, '', '&');
```

```
action=SALE&amount=2691&cardExpiryDate=1213&cardNumber=4929+4212+3460
+0821&countryCode=826&currencyCode=826&merchantID=101093&orderRef=Sig
nature+Test&transactionUnique=55f025addd3c2&type=1
```

Step 3 - Normalise all line endings in the url encoded string

Convert all CR NL, NL CR, CR character sequences to a single NL character.

```
$str = str_replace(array('%0D%0A', '%0A%0D', '%0D'), '%0A', $str);
```

```
action=SALE&amount=2691&cardExpiryDate=1213&cardNumber=4929+4212+3460
+0821&countryCode=826&currencyCode=826&merchantID=101093&orderRef=Sig
nature+Test&transactionUnique=55f025addd3c2&type=1
```

Step 4 - Append your signature key to the normalised string

The signature key is appended to the normalised string with no separator characters.

```
$str .= 'DontTellAnyone'
```

```
action=SALE&amount=2691&cardExpiryDate=1213&cardNumber=4929+4212+3460
+0821&countryCode=826&currencyCode=826&merchantID=101093&orderRef=Sig
nature+Test&transactionUnique=55f025addd3c2&type=1DontTellAnyone
```

Step 5 - Hash the string using the SHA-512 algorithm

The normalised string is hashed to a more compact value using the secure SHA-512 hashing algorithm[1].

```
$signature = hash('SHA512', $str);
```

```
da0acd2c404945365d0e7ae74ad32d57c561e9b942f6bdb7e3dda49a08fcddf74fe6a
f6b23b8481b8dc8895c12fc21c72c69d60f137fdf574720363e33d94097
```

Step 6 - Add the signature to the transaction form or post data

The signature should be sent as part of the transaction in a field called 'signature'.

```
<input type="hidden" name="signature" value="<?=$signature?>">
```

or

```
$tran['signature'] = $signature;
```

A-12 Transaction Life-cycle

Each transaction received by the Gateway follows a pre-determined life-cycle from receipt to completion. The stages in the life cycle are determined by the type of transaction and its success or failure at different stages in its life.

A-12.1 Authorise, Capture & Settlement

The key stages in the transactions life-cycle can be grouped into the Authorisation, Capture and Settlement stages as follows;

Authorisation

An authorisation places a hold on the transaction amount in the cardholder's issuing bank. No money actually changes hands yet. For example, let's say that you are going to ship a physical product from your website. First you authorise the amount of the transaction, then you ship the product. Only after the product is shipped do you capture the transaction.

Capture

A capture essentially marks a transaction as ready for settlement. As soon as the product is shipped, you can capture an amount up to the amount of the authorisation. Usually the full amount is captured. An example of a situation in which the whole amount is not captured might be if the Customer ordered multiple items and one of them is unavailable.

The Payment Gateway will normally automatically capture all authorisations as soon as they are approved freeing up you from having to do this.

However, it is usually more desirable to either delay the capture for a period of time or indefinitely. The **`captureDelay`** field can be used for this purpose and allow will allow you to state the number of days to delay any automatic capture or to never automatically capture. For more details on delayed capture refer to appendix A-8.

Settlement

Within 24 hours the Gateway will instruct your Acquirer to settle the transaction. The Acquirer then transfers the funds between the cardholder's and your accounts.

A-12.2 Transaction States

At any time during the transactions life cycle it is in one of a number of states as follows;

Received

The transaction has been received by the Gateway and stored away. This is the very first stage. The Gateway will examine the transaction and pass it on the next stage as appropriate.

Approved

The transaction has been sent to the Acquirer for authorisation and the Acquirer has approved it and is holding the cardholder's funds.

This is an intermediate state and follows the **received** state.

Verified

The transaction has been sent to the Acquirer for verification and the Acquirer has confirmed that the account is valid.

This is a terminal state and follows the **received** state. The transaction will never be settled and no funds will ever be transferred

Declined

The transaction has been sent to the Acquirer for authorisation and the Acquirer declined it.
The Acquirer will not normally give any reason for a decline and will not have held any funds.

The transaction has now completed its life-cycle and no more processing will be done on it.

This is a terminal state and follows the **received** state. The transaction will never be settled and no funds will ever be transferred. The transaction `responseCode` will be **5 (Declined)**.

Referred

The transaction has been sent to the Acquirer for authorisation and the Acquirer referred it for verbal approval.

The Merchant can choose not to seek verbal approval and treat these transactions the same as a normal 'declined' authorisation.

To seek verbal approval, the Merchant will need to phone the Acquirer and ask for an authorisation code. They will probably be asked for more information about the transaction and maybe required to gather other forms of identification from the Cardholder. If an authorisation code is provided then a new transaction can be sent to the Gateway specifying the **`xref`** of this transaction and the received **`authorisactionCode`**. This new request will not be sent for authorisation and will be in the 'approved' state ready for capture and settlement.

This is a terminal state and follows the **received** state. The transaction will never be settled and no funds will ever be transferred. The transaction **`responseCode`** will be **2 (Referred)**.

Reversed

The transaction was sent the Acquirer for authorisation and the Acquirer approved it however the transaction has been voided and the approval reversed. The Acquirer will have been asked to reverse any approval previously received effectively cancelling the authorisation and returning any held funds back to the Cardholder.

The gateway will reverse an authorisation if it declines the transaction post authorisation due to any AVS/CV checking. The PREAUTH action will also automatically reverse an authorisation before return

This is a terminal state and follows the **approved** state. The transaction will never be settled and no funds will ever be transferred

If the transaction was reversed due to AVS/CV2 checking, then the transaction **`responseCode`** will be **5 (AVS/CV2 Declined)**.

Captured

The transaction has been captured and the Acquirer will be asked to capture the approved held funds when the settling process next runs. The settling process normally runs each evening but the Acquirer may take up to 3 days to transfer the funds.

The **capture** state can either be entered automatically if the transaction requested an immediate or delayed capture or it can be manually requested by sending a CAPTURE request. You are free to change the amount to be captured to a value less than that initially approved by issuing one or more CAPTURE commands. Once captured there is no way to un-capture a transaction, if not explicitly cancelled, it will be sent for settlement at the next opportunity.

This is an intermediate state and follows the **approved** state.

Tendered

The transaction has been sent to the Acquirer for settlement by the settling process and is awaiting confirmation that it has been accepted.

At this point the transaction can no longer be cancelled or re-captured.

This is an intermediate state and follows the **captured** state.

Deferred

The transaction could not be settled due to some temporary problem such as a communications loss. It will be attempted again the next time the settling process runs – usually first thing the next day.

This is an intermediate state and follows the **tendered** state. It will normally be accompanied by a transaction response that indicates why the settlement process could not settle the transaction.

Accepted

The transaction has been accepted for settlement by the Acquirer. The held funds will be transferred between the Merchant and Cardholder in due course.

The transaction has now completed its life-cycle and no more processing will be done on it, unless it is subject to a rejection while the Acquirer is settling it.

This is a terminal state and follows the **tendered** state.

Rejected

The transaction has been rejected for settlement by the Acquirer. The held funds will not be transferred between the Merchant and Cardholder.

Few Acquirers inform the Gateway that they have rejected a transaction; they normally inform the Merchant directly. Therefore, a transaction may show as **accepted** even if was ultimately rejected or it may change from **accepted** to **rejected** if the Acquirer does inform the Gateway.

The transaction has now completed its life-cycle and no more processing will be done on it.

This is a terminal state and follows the **tendered** or **accepted** states. The transaction response will normally indicate the reason the transaction was rejected.

Cancelled

The transaction has been cancelled by the Merchant by sending a cancellation request to the Gateway either using the CANCEL action of via the Merchant Management System (MMS).

You can cancel any transaction that is not in a terminal state or in the 'tendered' state. Once cancelled any further processing that would have normally taken place will be halted. Cancelling a transaction may or may not release any funds held on the cardholder's card depending on support from the Acquirer and card scheme.

This is a terminal state and follows any non-terminal state that occurs before the transaction reaches the **tendered** state.

Finished

The transaction has finished and reached the end of its lifespan but did not reached one of the other terminal states. Usually this indicates a problem has occurred with the transaction that prevents it continuing with its normal life-cycle.

This is a terminal state can follow any other state. The transaction response will normally indicate the reason the transaction failed.

A-13 Transaction types

The Gateway only supports card not present (CNP) types of transactions, made where the cardholder does not or cannot physically present the card for a Merchant's visual examination at the time that an order is given and payment effected.

The type of transaction required is specified using the type request field when performing a new payment transaction.

A-13.1 E-commerce (ECOM)

E-commerce transactions are supported by the Gateway by using a transaction **`type`** of **1**. They are designed for Merchants who which to accept payments via a website, such as a shopping cart payment. E-commerce transactions can use advance fraud detection such as 3-D Secure.

Due to MasterCard stipulations the Gateway will not allow Maestro cards to be for new e-commerce transactions without the use of 3-D Secure.

A-13.2 Mail Order/Telephone Order (MOTO)

Mail Order/Telephone Order transactions are supported by the Gateway by using a transaction **`type`** of **2**. They are designed for Merchants who wish to build their own virtual terminal system to enter remote order details. You will need to ensure when processing such transactions, that their Acquirer understands the transaction is a MOTO transaction. The reason for this is because the Acquirer will have different requirements in order to classify a transaction as secure, e.g. 3-D Secure is often required for internet transactions, but impossible for MOTO transactions.

A-13.3 Continuous Authority (CA)

Continuous Authority transactions are supported by the Gateway by using a transaction **`type`** of **9**. They are designed for Merchants who wish to take full control of their subscription transactions. For further details on how to use Continuous Authority transactions please refer to Appendix A-15.2.

The Gateway offers a means of automating the taking of regular CA transactions using Recurring Transaction Agreements (RTA) as detailed in section 1.

A-14 Payment Tokenisation

All new transactions stored by the gateway are assigned a unique reference number which is referred to the cross reference and returned in the **`xref`** response field. This cross reference is displayed on the Merchant Management System (MMS) and used whenever a reference to a previous transaction is required.

The cross reference can be sent as part of a transaction request in the **`xref`** request field to tell the Gateway to perform an action on an existing transaction. This is normally for management actions such as **CANCEL** or **CAPTURE**.

The cross reference can also be sent with new transactions such as **PREAUTH**, **SALE**, and **REFUND** actions to request that the Gateway uses the values from the existing transactions if they have not been specified in the new request. For more information on how the existing values are used please refer to appendix A-16. This allows an existing transaction to be effectively repeated without you needing to know the original card number. The only exception to this is the card's security code (CVV) which, due to PCI:DSS restrictions, the Gateway cannot store this so it will have to be supplied in the new request (unless the new request is a Continuous Authority transaction, refer to appendix A-13.3).

The use of cross references to perform repeat transactions is referred to as Payment Tokenisation and should not be confused with Card Tokenisation which is a separate service offered by the Gateway and covered in a separate guide.

Refer to section 1 for details on how to instruct the Gateway to automatically repeat payment.

The way each action handles any supplied **`xref`** is as follows;

PREAUTH, SALE, REFUND, VERIFY requests

These requests will always create a new transaction.

The **`xref`** field can be provided to reference an existing transaction, which will be used to complete any missing fields in the current transaction; this previous transaction will not be modified. For more information on how the existing values are used please refer to appendix A-16. If the existing transaction cannot be found, then an error will be returned and recorded against the new transaction

The request is expected to contain any transaction information required.

The **`xref`** will only be used to complete any missing card and order details, preventing you from having to store card details.

REFUND_SALE requests

These requests will always create a new transaction.

The **`xref`** field can be provided to reference an existing transaction, which is going to be refunded. This existing transaction will be marked as have been fully or partially refunded and the amounts will be tallied to ensure you cannot refund more than the original amount of this existing transaction. If the existing transaction cannot be found, then an error will be returned and recorded against the new transaction.

The request is expected to contain any transaction information required.

The **`xref`** will not only be used to find the transaction to be refunded but that transaction will be used to complete any missing card and order details, preventing you from having to store card details.

CANCEL or CAPTURE requests

These requests will always modify an existing transaction.

The **`xref`** field must be provided to reference an existing transaction, which will be modified to the desired state. If the existing transaction cannot be found, then an error is returned but no record of the error will be recorded against any transaction.

The request should not contain any new transaction information any attempt to send any new transaction information will result in an error. The exception to this is that a CAPTURE request can send in a new lesser **`amount`** field when a lesser amount needs to be settled then was originally authorised.

QUERY requests

These requests will not create or modify any transaction.

The **`xref`** field must be provided to reference an existing transaction, which will be returned as if it had just been performed. If the existing transaction cannot be found, then an error is returned but no record of the error will be recorded against any transaction.

The request should not contain any new transaction information any attempt to send any new transaction information will result in an error.

SALE or REFUND Referred Authorisation requests

These will always create a new transaction.

The **`xref`** field must be provided to reference an existing transaction, which must be of the same request type and be in the **referred** state. A new transaction will be created based upon this transaction. If the existing

transaction cannot be found or is not in the **referred** state, then an error will be returned and recorded against the new transaction.

The new transaction will be put in the **approved** state and captured when specified by the existing or new transaction details. It will not be sent for authorisation again first.

The request may contain any new transaction but any card details or order amount must be the same as the existing transaction. Any attempt to send different card details or order details will result in an error.

NB: This usage is very similar to a normal SALE or REFUND request sent with an **`authorisationCode`** included; the difference being the **`xref`** must refer to an existing 'referred' transaction whose full details are used if required and not just an existing transaction whose card details are used if required. This means it is not possible to create a pre-authorised SALE or REFUND request and use a **`xref`** to mean use the card and order details from an existing transaction as a soon as the xref field is seen the Gateway assumes it is a **referred** transaction you wish to authorise.

A-15 Repeat Transactions

The Gateway only supports two main types of repeat transactions and the option for the Gateway to automatically take the repeat transactions on behalf of the Merchant.

Repeat transaction take advantage of the Payment Tokenisation feature of the Gateway as described in Appendix A-14 where each transaction is assigned a unique cross reference and allowing the details from a previous transaction to be used in a later transaction.

Refer to section **Error! Reference source not found.** for information on how the Gateway can be instructed to automatically take repeat payments depending on a pre-determined schedule.

A-15.1 Card On File Transactions

Transactions made using card details that have been previously captured and then stored 'on file' are termed 'Card On File' or 'COF' transactions. This is how most ad-hoc recurring/repeat transactions are performed using the **`xref`** field to refer to the card details stored on file during a previous transaction

A-15.1.1 Initial Transaction

The initial transaction can be any transaction that has successfully stored away valid credit card details and return a `xref` response field. The transaction need not have resulted in a successful authorisation but would normally be a successful VERIFY, PREAUTH or SALE request.

A-15.1.2 Repeat Transaction

The repeat transaction would send the `xref` returned by the initial transaction (or previous repeat transaction) as the `xref` request field. This transaction should use a `type` of **2** (MOTO) indicating it is a Cardholder not present transaction.

The repeat transaction would be a clone of the cross referenced transaction including any payment details with the exception of any new data provided in the repeat transaction. The `cloneFields` request field can also be used to control which fields in the cross referenced transaction are used in the repeat transaction (refer to Appendix A-16).

As the card CVV number is never stored then repeat transactions will either require the Cardholder to re-enter their CVV or the transaction has to be performed with no CVV. In such cases the Gateway will automatically suppress CVV checking however not all Acquirers will allow transactions to be performed with no CVV.

A-15.2 Continuous Payment Agreements

A Continuous Payment Authority (CPA), which is sometimes referred to as a recurring payment or a 'continuous payment transaction', is where the Cardholder gives a Merchant permission to regularly take money from their debit or credit card whenever they think they're owed money. Often payday loan companies, online DVD rental subscriptions, magazine subscriptions and gym memberships use this method of payment.

A-15.2.3 Initial Transaction

The initial transaction must be any successful VERIFY, PREAUTH or SALE request. If no payment is required at the same time then a Merchant must use a VERIFY request.

The initial transaction must be subject to the highest level of authentication supported. This would therefore mean that eCommerce transactions must use 3-D Secure when available.

To indicate that the initial transaction is the first in a Continuous Payment Authority then the type of agreement between the Merchant and the Cardholder must be specified using the `rtAgreementType` field.

The `rtAgreementType` can be one of the following values:

- **recurring** – this is used when each recurring payment may be for a variable or fixed amount and the agreement shall not have a specified end date.
- **instalment** – this is used when each recurring payment may be for a variable or fixed amount but the total of all the recurring payments will be for a fixed amount which shall be specified in the agreement with the cardholder. Therefore agreement has a specified end date and the total amount to be paid is known

A-15.2.4 Repeat Transaction

The repeat transaction would send the `xref` returned by the initial transaction (or previous repeat transaction) as the `xref` request field. This transaction must use a `type` of **9** (CA) indicating it is a Continuous Authority transaction.

The repeat transaction would be a clone of the cross referenced transaction including any payment details with the exception of any new data provided in the repeat transaction. The `cloneFields` request field can also be used to control which fields in the cross referenced transaction are used in the repeat transaction (refer to section).

As the card CVV number is never stored then repeat transactions will not require a card CVV to be supplied.

Acquirers insist that a separate acquiring account is used for any Continuous Authority payment in which case this would be associated with a different

Merchant Account. In such cases the initial transaction would be performed against your normal Merchant Account and the repeat transactions would be performed against your Continuous Authority Merchant Account.

It is the responsibility of the Merchant to regulate the transaction values and frequencies. Please be aware as a rule of thumb the banks expect Continuous Authority payments to be a predictable transaction amount on a regular or predictable frequency, any deviation from this can be viewed as an abuse of the Merchant's Continuous Authority acquiring account. You must also only ever process a Continuous Authority transaction on a card provided you have obtained full authorisation and authentication against that card via your normal Merchant Account.

Due to MasterCard stipulations the Gateway will not allow Maestro cards to be used with Continuous Authority transactions.

A-16 Transaction Cloning

If a new transaction request is received with the Cross Reference (**`xref`**) of an existing transaction, then the values of certain fields in the existing transaction will be used to initialise the new transaction where new values have not been provided in the new request. This copying of fields from a base transaction is termed '*transaction cloning*', the copied over value is termed the '*cloned value*'.

Appendix A-16.1 shows all the fields whose values can be copied over from the existing transaction. To easily allow for the addition of future fields the fields are grouped into logical groupings and each group is given a name (as show in brackets after the group title).

Certain groups of fields, such as address fields, can only be copied as a whole entity and any new value provided in the new request will prevent the whole group from being copied from the existing transaction. Please note line item data (`items`) cannot be merged.

By default the values of all the fields listed in Appendix A-16.1 are copied from the existing transaction where appropriate, however you can control exactly which fields are copied using the `cloneFields` field in the new request. The value of `cloneFields` should be a comma separated list of field names or group names that should be copied over. If, alternatively, you wish to specify a list of fields not to copy then prefix the list with a single exclamation mark (!).

Field Name	Mandatory?	Description
`cloneFields`	N	Comma separated list of field names or group names whose values should be cloned.

Examples

To copy over just the value of `customerName` and any values for the fields in the *customerAddressFields* group;

```
cloneFields="customerName, customerAddressFields"
```

To copy over the values of all supported fields apart from the value of `customerName` and `merchantName`;

```
cloneFields="!customerName,merchantName"
```

A-16.1 Cloned Fields

Transaction fields currently cloned are as follows:

A-16.1.5 Order Details Fields (*orderFields*)

- `type`
- `countryCode`
- `currencyCode`
- `amount`
- `grossAmount`
- `netAmount`
- `taxRate`
- `taxAmount`
- `taxReason`
- `discountAmount`
- `discountReason`
- `handlingAmount`
- `insuranceAmount`

A-16.1.6 Order Reference Fields (*orderRefFields*)

- `transactionUnique`
- `orderRef`
- `orderDate`

A-16.1.7 Card Fields (*cardFields*)

- `paymentMethod`
- `cardToken`
- `cardNumber`
- `cardExpiryDate`
- `cardExpiryMonth`
- `cardExpiryYear`
- `cardStartDate`
- `cardStartMonth`
- `cardStartYear`
- `cardIssueNumber`

A-16.1.8 Cardholder Fields (*cardholderFields*)

- `customerName`
- `customerAddress`
- `customerPostcode`
- `customerEmail`
- `customerPhone`

A-16.1.9 Purchase Fields (*purchaseFields*)

- `items`

A-16.1.10 Statement Narrative Fields (*narrativeFields*)

- `statementNarrative1`

- statementNarrative2

A-16.1.11 3D Secure Fields (*threedsFields*)

- threeDSRequired
- threeDSCheckRef

Please note: 3D Secure fields are only cloned if both the existing and new transaction are eCommerce transactions supporting 3-D Secure.

A-16.1.12 AVS/CV2 Fields (*avscv2Fields*)

- avscv2Required
- cv2CheckPref
- addressCheckPref
- postcodeCheckPref
- customerAddress
- customerPostcode

A-16.1.13 Merchant Email Notification Fields (*notifyFields*)

- notifyEmailRequired
- notifyEmail

A-16.1.14 Customer Receipt Fields (*cReceiptFields*)

- customerReceiptRequired
- customerEmail

A-16.1.15 Electronic Receipt Fields (*eReceiptFields*)

- eReceiptsRequired
- eReceiptsApiKey
- eReceiptsApiSecret
- eReceiptsStoreID
- eReceiptsCustomerRef
- eReceiptsReceiptRef
- eReceiptsReceiptData

A-16.1.16 Merchant Information Fields (*merchantFields*)

- merchantName
- merchantCompany
- merchantAddress*
- merchantTown*
- merchantCounty*
- merchantPostcode*
- merchantCountryCode*
- merchantPhone
- merchantMobile
- merchantFax
- merchantEmail
- merchantWebsite
- merchantData
- merchantOrderRef
- merchantCustomerRef
- merchantTaxRef

- merchantOriginalOrderRef
- merchantCategoryCode
- merchantType

A-16.1.17 Customer Information Fields (*customerFields*)

- customerName
- customerCompany
- customerAddress*
- customerTown*
- customerCounty*
- customerPostcode*
- customerCountryCode*
- customerPhone
- customerMobile
- customerFax
- customerEmail
- customerOrderRef
- customerMerchantRef
- customerTaxRef

A-16.1.18 Supplier Information Fields (*supplierFields*)

- supplierName
- supplierCompany
- supplierAddress*
- supplierTown*
- supplierCounty*
- supplierPostcode*
- supplierCountryCode*
- supplierPhone
- supplierMobile
- supplierFax
- supplierEmail

A-16.1.19 Receiver Information Fields (*receiverFields*)

- receiverName
- receiverCompany
- receiverAddress*
- receiverTown*
- receiverCounty*
- receiverPostcode*
- receiverCountryCode*
- receiverPhone
- receiverMobile
- receiverFax
- receiverEmail
- receiverAccountNo
- receiverDateOfBirth

A-16.1.20 Delivery Information Fields (*deliveryFields*)

- deliveryName
- deliveryCompany
- deliveryAddress*

- deliveryTown*
- deliveryCounty*
- deliveryPostcode*
- deliveryCountryCode*
- deliveryPhone
- deliveryMobile
- deliveryFax
- deliveryEmail

A-16.1.21 Shipping Information Fields (*shippingFields*)

- shippingMethod
- shippingTrackingRef
- shippingAmount
- shippingGrossAmount
- shippingNetAmount
- shippingTaxRate
- shippingTaxAmount
- shippingTaxReason
- shippingDiscountAmount
- shippingDiscountReason

A-16.1.22 MCC 6012 Additional Authorisation Data (*mcc6012Fields*)

- receiverName
- receiverPostcode
- receiverAccountNo
- receiverDateOfBirth

A-16.1.23 Payment Facilitator Data (*facilitatorFields*)

- subMerchantID
- facilitatorID
- facilitatorName

Please note: Payment facilitator fields are only cloned if the existing transaction uses the same ‘merchantID’ as the new transaction.

A-16.2 Cloned Groups

To easily allow for the future addition of new fields the existing fields are grouped into logic groupings each group is given a name (as show in brackets after the group title). It is recommended that this group name be used in any `cloneFields` value instead of listing all the fields separately.

A-16.2.1 Compound Groups

To help maintain transaction integrity certain groups of fields, such as address fields, can only be copied as a whole entity and any new value provided in the new request will prevent the whole group from being copied from the existing transaction.

These compound fields are marked with an asterisk in section 15.1 and can be referred to in `cloneFields` as logical groups using the following group names; *merchantAddressFields*, *customerAddressFields*, *deliveryAddressFields*, *supplierAddressFields* and *receiverAddressFields*

A-16.2.2 Line Item Data

Any line item data (`items`) is copied over in its entirety and there is no way to merge the line item from an existing transaction with any sent in a new transaction.

A-16.2.3 Amount Consistency

At present the Gateway does not validate that the various sub-amount fields such as `netAmount`, `grossAmount` etc. all add up to the actual requested `amount`. Therefore, these fields are currently not treated as a compound group.

If a new `amount` value is passed which is different to that in the existing transaction, then the following fields should be also be passed to they tally with the new amount.

- `grossAmount`
- `netAmount`
- `taxRate`
- `discountAmount`

A-17 Example Code

A-17.1 Example 3-D Secure SALE Transaction

The following example PHP code shows how to send a SALE transaction with support for 3-D Secure;

```php
<?PHP

// Signature key entered on MMS. The demo accounts is fixed to this value,
$key = 'Custom29Simple14Marker';

// Gateway URL
$url = 'https://gateway.fidelipay.co.uk/direct/';

// Request
$req = array(
   'merchantID' => '101094',
   'action' => 'SALE',
   'type' => 1,
   'countryCode' => 826,
   'currencyCode' => 826,
   'amount' => 1001,
   'cardNumber' => '4012001037141112',
   'cardExpiryMonth' => 12,
   'cardExpiryYear' => 15,
   'cardCVV' => '083',
   'customerName' => 'Test Customer',
   'customerEmail' => 'test@testcustomer.com',
   'customerAddress' => '16 Test Street',
   'customerPostCode' => 'TE15 5ST',
   'orderRef' => 'Test purchase',
   'transactionUnique' => (isset($_REQUEST['transactionUnique']) ?
$_REQUEST['transactionUnique'] : uniqid()),
   'threeDSMD' => (isset($_REQUEST['MD']) ? $_REQUEST['MD'] : null),
   'threeDSPaRes' => (isset($_REQUEST['PaRes']) ? $_REQUEST['PaRes'] : null),
   'threeDSPaReq' => (isset($_REQUEST['PaReq']) ? $_REQUEST['PaReq'] : null)
);

// Create the signature using the function called below.
$req['signature'] = createSignature($req, $key);

// Initiate and set curl options to post to the gateway
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($req));
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Send the request and parse the response
parse_str(curl_exec($ch), $res);

// Close the connection to the gateway
curl_close($ch);
```

```
// Extract the return signature as this isn't hashed
$signature = null;
if (isset($res['signature'])) {
   $signature = $res['signature'];
   unset($res['signature']);
}

// Check the return signature
if (!$signature || $signature !== createSignature($res, $key)) {
   // You should exit gracefully
   die('Sorry, the signature check failed');
}

// Check the response code
if ($res['responseCode'] == 65802) {

   // Send details to 3D Secure ACS and the return here to repeat request
   $pageUrl = (@$_SERVER['HTTPS'] == 'on') ? 'https://' : 'http://';
   if ($_SERVER['SERVER_PORT'] != '80') {
      $pageUrl .= $_SERVER['SERVER_NAME'] . ':' . $_SERVER['SERVER_PORT'] .
$_SERVER['REQUEST_URI'];
   } else {
      $pageUrl .= $_SERVER['SERVER_NAME'] . $_SERVER['REQUEST_URI'];
   }

   echo "
<p>Your transaction requires 3D Secure Authentication</p>
<form action=\"" . htmlentities($res['threeDSACSURL']) . "\"method=\"post\">
<input type=\"hidden\" name=\"MD\" value=\"" . htmlentities($res['threeDSMD']) . "\">
<input type=\"hidden\" name=\"PaReq\" value=\"" . htmlentities($res['threeDSPaReq']) .
"\">
<input type=\"hidden\" name=\"TermUrl\" value=\"" . htmlentities($pageUrl) . "\">
<input type=\"submit\" value=\"Continue\">
</form>
";

} else if ($res['responseCode'] === "0") {
   echo "<p>Thank you for your payment.</p>";
} else {
   echo "<p>Failed to take payment: " . htmlentities($res['responseMessage']) .
"</p>";
}

// Function to create a message signature
function createSignature(array $data, $key) {
   // Sort by field name
   ksort($data);

   // Create the URL encoded signature string
   $ret = http_build_query($data, '', '&');

   // Normalise all line endings (CRNL|NLCR|NL|CR) to just NL (%0A)
   $ret = str_replace(array('%0D%0A', '%0A%0D', '%0D'), '%0A', $ret);

   // Hash the signature string and the key together
   return hash('SHA512', $ret . $key);
}

?>
```

A-17.2 Example None 3-D Secure Sale Transaction

The following sample PHP code shows how to send a SALE transaction without support for 3-D Secure;

```php
<?PHP

// Signature key entered on MMS. The demo accounts is fixed to this value,
$key = 'Custom29Simple14Marker';

// Gateway URL
$url = 'https://gateway.fidelipay.co.uk/direct/';

// Request
$req = array(
   'merchantID' => '101093',
   'action' => 'SALE',
   'type' => 1,
   'countryCode' => 826,
   'currencyCode' => 826,
   'amount' => 1001,
   'cardNumber' => '4012001037141112',
   'cardExpiryMonth' => 12,
   'cardExpiryYear' => 15,
   'cardCVV' => '083',
   'customerName' => 'Test Customer',
   'customerEmail' => 'test@testcustomer.com',
   'customerPhone' => '+44 (0) 123 45 67 890',
   'customerAddress' => '16 Test Street',
   'customerPostCode' => 'TE15 5ST',
   'orderRef' => 'Test purchase',
   'transactionUnique' => uniqid(),
);

// Create the signature using the function called below.
$req['signature'] = createSignature($req, $key);

// Initiate and set curl options to post to the gateway
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($req));
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Send the request and parse the response
parse_str(curl_exec($ch), $res);

// Close the connection to the gateway
curl_close($ch);

// Extract the return signature as this isn't hashed
$signature = null;
if (isset($res['signature'])) {
   $signature = $res['signature'];
   unset($res['signature']);
}
```

```php
// Check the return signature
if (!$signature || $signature !== createSignature($res, $key)) {
   // You should exit gracefully
   die('Sorry, the signature check failed');
}

// Check the response code
if ($res['responseCode'] === "0") {
   echo "<p>Thank you for your payment.</p>";
} else {
   echo "<p>Failed to take payment: " . htmlentities($res['responseMessage']) .
"</p>";
}

// Function to create a message signature
function createSignature(array $data, $key) {
   // Sort by field name
   ksort($data);

   // Create the URL encoded signature string
   $ret = http_build_query($data, '', '&');

   // Normalise all line endings (CRNL|NLCR|NL|CR) to just NL (%0A)
   $ret = str_replace(array('%0D%0A', '%0A%0D', '%0D'), '%0A', $ret);

   // Hash the signature string and the key together
   return hash('SHA512', $ret . $key);
}

?>
```

A-18 Frequently Asked Questions

1. I'm getting Invalid Credentials. What do I do?

- Check your Merchant ID in your integration is correct. Our Gateway Merchant IDs typically begin with 1 and are currently 6 digits long, e.g. 101093.

2. I'm getting an invalid signature error message. How do I fix it?

- Check you are using the correct method for calculating the signature and the correct secret signature key for the Merchant Account used.

- Make sure you are not using an image form submit button as that will add fields to the post which cannot be removed and will render the signature useless.

Refer to appendix A-11 for a step by step guide to creating a signature with same values which you can using your own signature generation routing to see if it produced the same value as ours. This test step by step generator is available on our website, just click on the link before and follow the instructions.

https://gateway.fidelipay.co.uk/devtools/sigtest.php

3. I have more than one Merchant ID - how do I use more than one?

- You have a couple options here. You can setup separate integrations for each MID, which can be a bit inconvenient. Your other option is to request they are connected together. Please contact our support team to get your MIDs connected and you will then only need to use one.

4. I receive a 'Bad Testcard Usage' error message. Why?

- If you receive this error message you are using test cards on a live Merchant ID. Please only use live cards on live Merchant IDs. Our test cards will only work on the test Merchant ID provided when you sign up with us.