HOSTED

INTEGRATION GUIDE

Version: 9.16

1 Hosted HTTP Integration

1.1 About This Guide

The Hosted HTTP Integration method requires the Merchant (or the Merchant's web developer) to have knowledge of server side scripting languages (e.g. PHP, ASP etc.).

Unlike the Direct method, the Merchant's website does not need to have a SSL Certificate, and PCI compliance becomes more straightforward.

If you wish to process card details on your own website, or style the payment pages of your website, you need to use the Direct integration method.

This guide provides the information required to integrate with the Payment Gateway and gives a very basic example of code for doing so. It is expected that the Merchant, or the Merchant's developers, 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 UTP Merchant Services Ltd on 0118 953 0953 or via email to support@universaltp.com.

1.2 Integration Disclaimer

UTP 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. UTP is not responsible or liable for any Merchant or Third Party integration.

1.3 Pre-Requisites

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

UTP 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.universaltp.com/paymentform/

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 the use the test cards to run a test transaction.

For non-3-D-Secure testing use Merchant ID **101073**
For 3-D Secure Testing use Merchant ID **101074**

1.4 Authorise, Capture & Settle

A transaction sent to the Payment Gateway goes through various stages during its life time. The notable stages are shown below;

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 the Merchant is going to ship a physical product from their website. First they authorise the amount of the transaction. Then they ship the product. Only after the product is shipped does the Merchant capture the transaction.

Capture

A capture essentially marks a transaction as ready for settlement. As soon as the product is shipped, the Merchant 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.

Settlement

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

A transaction can normally be cancelled any time between a success authorisation and settlement such cancellation. Depending on the Merchant's Acquirer a cancellation may request that the Acquirer void the authorisation and remove the hold on the cardholder's funds.

The Payment Gateway will normally automatically capture all authorisations as soon as they are approved freeing up the Merchant 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 the Merchant 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-7.

1.5 Request Actions

The Hosted Integration allows various standard actions to be performed via providing different values for the **`action`** field as follows;

1.5.1 PREAUTH

This will create a new transaction and attempt to seek authorisation for a sale from the Acquirer. The transaction will not be captured and settled. In addition the authorisation will be voided (where possible) so funds are not reserved on the cardholder's account. This transaction type can be used to check whether funds are available (at that time) and that the account is valid. Any xref response can be used in lieu of the card details in a subsequent transaction – it therefore provides a very simple card tokenisation facility.

Note: Any authorisation approved using a PREAUTH transaction is voided automatically and thus must be collected by doing a new authorisation using the SALE action. If the 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 recommended to only PREAUTH small amounts such as £1 to mainly check account validity.

To indicate that a PREAUTH was performed a successful transaction will have its state set to 'reversed' meaning that it cannot be later captured and a new SALE must be made instead.

It is recommended that Merchants use the VERIFY action instead where supported by their Acquirer.

1.5.2 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 captured and settled and how many days to wait between the authorisation and settlement. For more details on delayed capture refer to Appendix A-7.

1.5.3 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 and will not be captured or settled. The transaction **`amount`** must always be zero.

This transaction type is the preferred method for validating an account but won't validate if sufficient funds aren't present. It should be used, when supported, instead of doing small PREAUTH transactions aimed at checking account validation.

This page is intentionally left blank.

1.6 Security

You can add additional security to a transaction by setting up one or more security features such as allowed IP addresses, password authentication, message signing and callback URLs.

1.6.1 Allowed IP addresses

You can configure a list of allowed IP addresses using the Merchant Management System (MMS) merchant preferences section. If a Hosted integration transaction is received from an address other than those configured then it will be rejected with a **`responseCode`** of 65540 (PERMISSION DENIED). Separate lists can be configured for standard transactions such as PREAUTH, SALE and VERIFY and advanced or management transactions such as REFUND, CAPTURE and CANCEL etc.

1.6.2 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. If sent password does not match the one entered in the MMS then the transaction with fail with a **`responseCode`** of **65539 (INVALID CREDENTIALS)**.

Use of a password is discouraged in any integrations where the transaction is posted from a form on the Merchant's website as the password would appear in plain text in the HTML source code.

1.6.3 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 fail with a **`responseCode`** of **64439 (INVALID SIGNATURE)**.

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.

See Appendix 12.6A-11 for information on how to create the hash.

1.6.4 Callback URL

You can request that the Gateway sends the results of each transaction to URL a specified in the **`callbackURL`** field. Then each transaction result will be POSTed 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.

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

1.7 Payment Tokenization

All new transactions stored by the Gateway are assigned a unique reference number which is referred to the cross reference (xref) 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 also be sent as part of a transaction request in the xref request field, this is normally for management actions such as CANCEL, CAPTURE etc. to tell the Gateway which transaction to act on.

However it can also be sent with new transactions such as PREAUTH, SALE, and REFUND etc. to request that it uses the existing transactions details to complete the new one. This allows a transaction to be effectively repeated without the Merchant needing to know the original card numbers. However the CV2 value is never stored by the Gateway so the repeat transaction cannot perform CV2 checks (Refer to section 3).

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

Refer to Appendix A-8 for more information on how each request handles the receipt of a cross reference etc.

2 Basic Transactions

2.1 Implementation

The Merchant will need to send the request details to the integration URL via an HTTP POST request.

The transaction should be sent as URL encoded Name=Value fields separated by '&' characters. The response will be received in the same format.

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

The request needs to be sent from a web browser as the response will be a HTML Hosted checkout page requesting the Customer enter their card details etc. The normal way to achieve this is to send the request data as hidden form fields as per the example code provided in Appendix A-11. The browser will then automatically encode the request correctly as per the `application/x-www-form-urlencoded` format.

Once the checkout page has been complete the Customers browser will be automatically redirected to the URL provided via the **`redirectURL`** field. The response will the returned to this page using a HTTP POST request.

If a callback URL is provided via the **`callbackURL`** field then any response will also be sent to the callback using a HTTP POST request.

Note: 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.

Please note that the field names are cAsE sEnSiTiVe.

2.2 Request Fields

Field Name	Mandatory?	Description
`merchantID`	Yes	Your UTP Merchant ID.
`merchantPwd`	No[1]	Any password used to secure this account. Refer to section 9 for details.
`signature`	Yes[2]	Any hash used to sign the transaction request. Refer to section 9 for details.
`redirectURL`	Yes	The URL to which the Customer will be redirected and the transaction result will be POSTed when the checkout is completed.
`amount`	Yes	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.**
`action`	Yes	The action requested. Refer to section 1.4 for details on the various actions allowed. Possible values are: **PREAUTH, SALE, REFUND, REFUND_SALE, VERIFY, CAPTURE, CANCEL, QUERY**
`type`	Yes	The type of transaction. Possible values are: **1** – Ecommerce, **2** - Mail Order, **9** – Continuous Authority
`countryCode`	Yes	ISO standard alpha or numeric country code for the Merchant's location.
`currencyCode`	Yes	ISO standard alpha or numeric currency code for this transaction. You may only use currencies that are enabled for your Merchant account.
`transactionUnique`	No	A unique identifier for this transaction. This should be set by your website or shopping cart. This is an added security feature to combat transaction spoofing.
`orderRef`	No	This text field allows you to describe the order or provide an invoice number/reference number for the

		Merchant's records.
captureDelay	No	Number of days to wait between authorisation of a payment and subsequent settlement. Refer to Appendix A-7 for further details.
callbackURL	No	A non-public URL which will receive a copy of the transaction result by POST. Refer to section 9 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.

2.3 Response Fields

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 detail or any error that occurred. For a full list of error codes please refer to the table in Appendix A.
responseMessage	Always	The message received from the Acquiring bank, or any error message.
xref	Always	The Merchant may store the cross reference for repeat transactions. Refer to section 1.7
transactionUnique	If supplied	Any value supplied in the initial request.
amountReceived	On success	The amount of the transaction. This field used in conjunction with **transactionUnique** can help provide a measure of security.
transactionID	Always	A unique ID assigned by the Gateway.
orderRef	If supplied	Any value supplied in the initial request.
cardNumberMask	Always	Card number masked so only the last 4 digits are visible.
cardTypeCode	Always	The code of card used. See appendix A-2 for a full list.
cardType	Always	The description of the card used. See Appendix A-2 for a full list.

Note: the response is POSTed to any URL provided by the mandatory **redirectURL** and optional **callbackURL**.

3 AVS/CV2 Checking

3.1 Background

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

These fraud prevention checks are performed by the Merchant's Acquirer on application transaction. The Merchant can choose how to act on the outcome of the check (or even to ignore them altogether).

3.1.1 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.

3.1.2 CV2 Checking

CV2, CVV, or Card Verification Value is a 3 or 4 digit security code – commonly found on the reverse of cards on the signature strip. The check verifies the code is the correct one for the card used.

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 3.3 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)**.

3.2 Benefits & Limitations

3.2.1 Benefits

- Instant: The results are available immediately and returned as part of the transaction.

- Flexible: The checks can be managed independently allowing the Merchant the upmost control over how the results are used.

- Automatic: The checks can be configured by the Merchant to automatically decline transaction where required.

3.2.2 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.

3.3 Request Fields

Field Name	Mandatory?	Description
`customerAddress`	Yes[3]	The Customer or cardholder's address. For AVS checking this must be the registered billing address of the card.
`customerPostCode`	Yes[4]	The Customer or cardholder's postcode. For AVS checking this must be the registered billing postcode of the card.
`cardCVV`	Yes[5]	The Customer's card CVV number. This is a three (or four for American Express) digit numeric printed on the back of the card. **Numeric values only**
`cv2CheckPref`	No[6]	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 values**: 'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
`addressCheckPref`	No[6]	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 values**: 'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
`postcodeCheckPref`	No[6]	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 values: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**

[3] Mandatory if AVS address checking is required
[4] Mandatory if AVS postcode checking is required
[5] Mandatory if CV2 checking is required
[6] If the value is not supplied than the default account preferences will be used.

3.4 Response Fields

Field Name	Returned?	Description
avscv2ResponseCode	Optional	The result of the AVS/CV2 check. Please see Appendix A-4 for a full list of possible responses.
avscv2ResponseMessage	Optional	The message received from the Acquiring bank, or any error message with regards to the AVS/CV2 check. Please see Appendix A-4 for a full list of possible responses.
avscv2AuthEntity	Optional	Textual description of the AVS/CV2 authorizing entity as described in Appendix A-3. Possible values are: **'not known', 'merchant host', 'acquirer host', 'card scheme', 'issuer'**
cv2Check	Optional	Textual description of the AVS/CV2 check as described in Appendix A-4. Possible values are: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
cv2CheckPref	Optional[1]	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 values: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
addressCheck	Optional	Textual description of the AVS/CV2 address check as described in Appendix A-4. Possible values are: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
addressCheckPref	Optional[1]	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 values: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
postcodeCheck	Optional	Textual description of the AVS/CV2 postcode

		check as described in Appendix A-4. Possible values are: **'not known', 'not checked', 'matched', 'not matched', 'partially matched'**
postcodeCheckPref	Optional[1]	List of **postcodeCheck** 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', 'matched', 'not matched', 'partially matched'**

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

4 3-D Secure Authentication

4.1 Background

3-D Secure authentication is an additional fraud prevention scheme that is available to all Merchant's 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 implement 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.

4.2 Benefits & Limitations

4.2.1 Benefits

- Instant: The results are available immediately and returned as part of the transaction.

- Flexible: The checks can be managed independently allowing the Merchant the upmost control over how the results are used.

- Automatic: The checks can be configured by the Merchant 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 chargebacks as the liability is assumed. Note: Merchants will need to confirm with their Acquirer for 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).

4.2.2 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.

4.3 Implementation

If your merchant account is enrolled with 3-D Secure, the default Gateway Hosted form will automatically attempt to display the 3-D Secure authentication page for the Customers bank.

The 3-D Secure authentication form is designed and controlled by the Customers bank but the Merchant can change their name and website address that is displayed on the form by sending the **`merchantName`** and/or **`merchantWebsite`** request fields.

4.4 Request Fields

Field Name	Mandatory?	Description
`merchantName`	No[1]	Merchant name to use on 3-D Secure form
`merchantWebsite`	No[1]	Merchant website to use on 3-D Secure form
`threeDSRequired`	No[2]	Is 3-D Secure required for this transaction Possible values are: **N** – 3-D Secure is not required **Y** – 3-D Secure is required (abort if not available)
`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', 'matched', 'not matched', 'partially matched'**

[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**

This page is intentionally left blank

4.5 Response Fields

When a 3-D Secure transaction is processed then the following additional fields may be returned.

Field Name	Returned?	Description
`threeDSEnabled`	Yes	Is 3-D Secure enabled for the merchant account? Possible values are: **N** – the Merchant is not 3DS enabled **Y** – the Merchant is 3DS enabled
`threeDSRequired`	Yes	Was 3-D Secure required for this transaction? Possible values are: **N** – 3DS was not required **Y** – 3DS was required
`threeDSCheckPref`	Yes	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', 'matched', 'not matched', 'partially matched'**
`threeDSEnrolled`	Yes	The 3-D Secure enrolment status for the credit card. Possible values are: **Y** - Enrolled **N** - Not Enrolled **U** - Unable To Verify **E** - Error Verifying Enrolment Refer to Appendix 12.6A-4 for further information.
`threeDSAuthenticated`	Yes	The 3-D Secure authentication status for the credit card. Possible values are: **Y** - Authentication Successful **N** - Not Authenticated **U** - Unable To Authenticate **A** - Attempted Authentication **E** - Error Checking Authentication Refer to Appendix 12.6A-4 for further information.

Field Name	Returned?	Description
threeDSPaReq	Yes	Payer Authentication Request (PaReq) that is sent to the Access Control Server (ACS) in order to verify the 3-D Secure status of the credit card.
threeDSPaRes	Yes	Payer Authentication Response (PaRes) that is returned from the Access Control Server (ACS) determining the 3-D Secure status of the credit card.
threeDSACSURL	Yes	The URL of the Access Control Server (ACS) to which the Payer Authentication Request (PaReq) should be sent.
threeDSECI	Yes	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	Yes	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	Yes	This contains the one digit value which indicates the algorithm used by the Access Control Server (ACS) to generate the CAVV. Valid algorithms include (amongst others): **0** - HMAC **1** - CVV **2** - CVV with ATN The data contained within this property is only valid if the **threeDSAuthenticated** value is **Y** or **A**.
threeDSXID	Yes	A unique identifier for the transaction as used in the 3-D Secure process.

Field Name	Returned?	Description
threeDSErrorCode	Yes	Any error response code returned by the 3-D Secure Access Control Server (ACS) should there be an error in determining the card's 3-D Secure status.
threeDSErrorDescription	Yes	Any error response description returned by the 3-D Secure Access Control Server (ACS) should there be an error in determining the card's 3-D Secure status.
threeDSVETimestamp	Yes	The time the card was checked for 3-D Secure enrolment.
threeDSCATimestamp	Yes	The time the card was checked for 3-D Secure authentication.

This page is intentionally left blank

5 VISA MCC6012 Merchants

5.1 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 Merchant will be told by their Acquirer if they need to send this information. .

5.1.1 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.

5.1.2 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.

5.2 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.

6 Billing Descriptor

6.1 Background

The Billing Descriptor is how the Merchant's details appear on the cardholder's statement 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.

6.1.1 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.

6.1.2 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 Merchants 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.

6.7 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 information

7 Receipts & Notifications

7.1 Background

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

7.1.1 Customer Email Receipts

The cardholder (Customer) can be automatically emailed a transaction receipt each time a transaction is processed by the Gateway. Receipts are only sent for transactions which are either approved, declined or referred by the Acquirer.

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. The default Hosted form will force the Customer to provide an email address if the **`customerEmailRequired`** request field is sent as **`Y`**.

7.1.2 Merchant Email Notifications

The Merchant can be automatically emailed a transaction notification each time a transaction is processed by the Gateway. Notifications are only sent for transactions which are either approved, declined or referred by the Acquirer.

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.

7.1.3 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. Electronic receipts are only stored for transactions which are approved by the Acquirer.

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.

Electronic receipts require the Merchant to supply a unique Customer reference (using the **`eReceiptsCustomerRef`** field) or, alternatively, the Customer to provide their email address (using the **`customerEmail`** field) or 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 **Error! Reference source not found.**, for information on which fields are used to build the electronic receipt refer to section 7.2 below.

7.2 Request Fields

7.2.1 General Fields

Field Name	Mandatory?	Description
customerReceiptsReqired	No[1]	Send a Customer receipt if possible
customerEmail	No	Customer's email address.
notifyEmailRequired	No[1]	Send a notification email if possible.
notifyEmail	No[1]	Merchant's notification email address.
eReceiptsRequired	No[1]	Send receipt data to eReceipts™ system.
eReceiptsStoreID	No[1]	Merchant's eReceipts™ store identifier.
eReceiptsCustomerRef	No[2]	Merchant's eReceipts™ Customer reference.
eReceiptsApiKey	No[1]	Merchant's eReceipts™ API key.
eReceiptsApiSecret	No[1]	Merchant's eReceipts™ API secret.
eReceiptsReceiptRef	No	Merchant's eReceipts™ receipt reference.
eReceiptsReceiptData	No[3]	Complete eReceipt™ data

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

7.2.2 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	Reason for discount for *XX*th item purchased.
item*XX*ProductCode	No	Product code for *XX*th item purchased.
item*XX*CommodityCode	No	Commodity code for *XXth* 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.

7.3 Response Fields

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

Field Name	Description
customerReceiptsResponseCode	Result of sending email to Customer.
customerReceiptsResponseMessage	Description of above response code.
notifyEmailResponseCode	Result of sending email to Merchant.
notifyEmailResponseMessage	Description of above response code.
eReceiptsEnabled	Is the eReceipts system enabled (Y|N).
eReceiptsResponseCode	Result of sending details to eReceipts™.
eReceiptsResponseMessage	Description of above response code.
eReceiptsReceiptRef	Unique eReceipt™ reference.

8 Purchase Data

8.1 Background

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

The Gateway provides a number of fields which the Merchant 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.

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

8.1.2 Purchase Orders

These fields along with other advanced fields as detailed in section 12 can be used by the Merchant 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.

8.2 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*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*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*DiscountAmount[2]	No	Total discount for *XX*th item purchased.
item*XX*DiscountReason[2]	No	Reason for discount 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 *XXth* 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 **`itemXX`** 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.

9 Recurring Transaction Agreements

9.1 Background

The Gateway makes it easy to do repeating payments either ad-hoc or to a pre-configured schedule.

Ad-hoc agreements can be performed by providing the cross reference to an existing transaction instead of (or in addition to) the card and order details when requesting a new transaction.

Scheduled recurring transactions are configured be setting up a Recurring Transaction Agreement. This is an agreement with the Gateway telling it when and how to automatically take recurring payments.

Repeat payment can be done as either ‘Card On File’ or ‘Continuous Authority’ transactions.

For more information, refer to our **Recurring Transactions** guide.

9.1.1 Card On File transactions (COF)

Transactions made using card details that have been previous capture and then stored ‘on file’ are termed ‘Card On File’ transactions. This is how most ad-hoc repeat transactions are performed using the **`xref`** field to refer to the card details stored on file during a previous transaction. As the card CVV number is never stored then such 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.

9.1.2 Continuous Payment Authority (CPA) transactions

A Continuous Payment Authority (CPA), which is sometimes referred to as a recurring payment or a ‘continuous payment transaction’, is where the Cardholder gives the 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, porn websites, magazine subscriptions and gym memberships use this method of payment.

Often the Acquirer will require the Merchant to use a specific Merchant account different to their normal account for CPA transactions.

9.2 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)
rtType	No	Recurring transaction type (ECOM, MOTO, CA)
rtUnique	No	'transctionUnique' value for recurring transaction
rtMerchantID	No	Merchant ID to make recurring payments on
rtStartDate	No	Start date of agreement (default to transaction date)
rtInitialDate	No	Date of initial payment (YYYY-MM-DD HH:MM:SS)
rtInitialAmount	No	Amount of initial payment (default to cycle amount)
rtFinalDate	No	Date of final payment (YYYY-MM-DD HH:MM:SS)
rtFinalAmount	No	Amount of final payment (default to cycle amount)
rtCycleAmount	No	Amount per cycle (default to initial amount)
rtCycleDuration	Yes	Cycle duration
rtCycleDurationUnit	Yes	Cycle duration unit. One of 'day', 'week', 'month', 'year'
rtCycleCount	No	Period (as number of cycles)
rtMerchantData	No	Free format Merchant data field

10 Duplicate Transaction Checking

10.1 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).

10.2 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.

10.3 Request Fields

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

11 Custom Request Data

The Merchant 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†.

The Merchant can put extra information that should be stored into the **`merchantData`** field. Associative data can be serialised using the notation **`merchantData[name]=value`**.

†*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.*

11.6 Request Fields

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

12 Advanced Integration Fields

The Gateway provides a number of fields which the Merchant 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.

12.1 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 cards registered address.
`customerPostcode`	No[1]	Cardholder's cards registered postcode.
`customerTown`	No	Cardholder's cards registered town/city.
`customerCounty`	No	Cardholder's cards registered county/province.
`customerCountryCode`	No	Cardholder's cards registered country. **Valid ISO 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

12.2 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 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).

12.3 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 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.

12.4 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 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.

12.5 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 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.
receiverDataOfBirth	No	Receiver's date of birth.

12.6 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.

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.

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	Reserved for future use. Contact customer support if this error occurs
65542	Request Mismatch: fields sent while completing a request do not match initially requested values. Usually due to sending different card details when completing a 3-D Secure transaction to those used to authorise the transaction
65543	Request Ambiguous: request could be misinterpreted due to inclusion of mutually exclusive fields
65544	Request Malformed: couldn't parse the request data
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 communications error
65552	Internal error
65553	Encryption error
65554	Duplicate request. Refer to Section 10
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 seconday 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 cannot 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
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)
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
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 10.
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
66344	Invalid Authorisation Code
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
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 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.

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-3 AVS / CV2 Check Response

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-4 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 authorisation with ECI 7. The Acquirer/Merchant retains liability if the cardholder later disputes making the purchase.

E - Error Verify Enrolment: The Gateway system 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 authorisation 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 authorisation processing.

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

A-5 Standard Hosted Form Options

The Hosted integration allows a few fields to be sent in the request that can customise the appearance and/or behaviour of the standard Hosted checkout form as follows;

Field Name	Mandatory?	Description
`cardNumber`	No	Default value to display in the card number field. **It is highly recommended that this field never be sent!**
`cardCVV`	No	Default value to display in the CVV field.
`cardExpiryMonth`	No	Default value to display as the card expiry month.
`cardExpiryYear`	No	Default value to display as the card expiry year.
`customerName`	No	Default value to display in the cardholder's name field.
`customerAddress`	No	Default value to display in the cardholder's postal address field.
`customerPostcode`	No	Default value to display in the cardholder's postcode field.
`customerEmail`	No	Default value to display in the cardholder's email address field.
`customerPhone`	No	Default value to display in the cardholder's phone number field.
`cardCVVMandatory`	No	Force a CVV to be entered.
`customerAddressMandatory`	No	Force a postal address to be entered.
`cusotmerPostcodeMandatory`	No	Force a postcode to be entered.
`customerEmailMandatory`	No	Force an email address to be entered.
`customerPhoneMandatory`	No	Force a phone number to be entered.

A-6 Request Checking Only

Sometimes the Merchant may wish to submit a request via the Hosted HTTP interface method in order for it to be validated only and not processed or sent to the financial institution for honouring. 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-7 Capture Delay

Capture Delay enables you 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-8 Cross References

Each transaction is assigned a unique cross reference which is returned in the **`xref`** response field, the value can then be passed to a subsequent in its **`xref`** request field.

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

AUTH or SALE request

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 card fields in the current transaction; this previous transaction will not be modified.

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 the Merchant from having to store card details.

REFUND_SALE request

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 the Merchant from having to store card details.

CANCEL or CAPTURE request

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 previous transaction cannot be found then an error is returned but no record of the error will 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 request

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 previous 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 request

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 authorisation code 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-9 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 the Merchant's servers and never left for the client browser to do in JavaScript as this would mean revealing the Merchant's 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' => '101073',
  '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' => '101073', '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=101073&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("\r\n", "\n\r", "\r"), "\n", $str);
```

```
action=SALE&amount=2691&cardExpiryDate=1213&cardNumber=4929+4212+3460
+0821&countryCode=826&currencyCode=826&merchantID=101073&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=101073&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-10 Example Signature Creation Code

The following example PHP code shows how to create the transaction signature;

```php
<?PHP
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
   $ret = hash('SHA512', $ret . $key);

   return $ret;
}
?>
```

A-11 Example Code

The following example PHP code shows how to send a SALE transaction; please use the createSignature function above to get the below code to work.

```php
<?PHP

// Signature key entered on MMS. The demo accounts is fixed,
// but merchant accounts can be updated from the MMS .
$key = 'Train37There28Metal';

// Gateway URL
$url = 'https://gateway.universaltp.com/paymentform/';

if (!isset($_POST['responseCode'])) {
   // Send request to gateway

   // Request
   $req = array(
      'merchantID' => '101073',
      'action' => 'SALE',
      'type' => 1,
      'countryCode' => 826,
      'currencyCode' => 826,
      'amount' => 1001,
      'orderRef' => 'Test purchase',
      'transactionUnique' => uniqid(),
      'redirectURL' => ($_SERVER['HTTPS'] == 'on' ? 'https' : 'http') . '://' .
$_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'],
   );

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

   echo '<form action="' . htmlentities($url) . '" method="post">' . PHP_EOL;

    foreach ($req as $field => $value) {
      echo '    <input type="hidden" name="' . $field . '" value="' .
htmlentities($value) . '">' . PHP_EOL;
   }

    echo '    <input type="submit" value="Pay Now">' . PHP_EOL;
   echo '</form>' . PHP_EOL;

} else {
```

```
    // Handle the response posted back
    $res = $_POST;

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

}

?>
```

This page is intentionally left blank

This page is intentionally left blank

A-12 Test Cards

Note: DON'T USE THESE TEST CARDS ON LIVE MIDS. THEY ARE FOR TEST MIDS 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

The authorisation response is dependent on the transaction amount:

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

Visa Credit

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

UK Maestro

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

Electron

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

American Express

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

Diners Club

Card Number
36432685260294

A-13 3-D Secure Test Cards

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 Number	Address	Postcode	Amount	Test Scenario
4909630000000008				£12.01	Card range not participating
4012010000000000 009				£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

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 fallback password is dog33cat. Use Visa's 3-D Secure test cards if these are not behaving as expected.

Card Number	CVV Number	Address	Postcode	Amount	Test Scenario
5033961989000000818	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-14 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. 101073.

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.

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.