DIRECT

INTEGRATION GUIDE

Version: 9.15

1 Direct HTTP Integration

1.1 About This Guide

The Cardstream Direct 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 Hosted method, the Merchant's website must have a SSL Certificate, and maybe required to be PCI compliant.

If you wish to process card details on your own website, or style the payment pages of your website, you either need to use the Direct integration method or request a Custom Hosted Form for your business.

1.2 Cardstream Integration Disclaimer

Cardstream provides all integration documentation necessary for enabling merchant clients 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. Cardstream is not responsible or liable for any Merchant or Third Party integration.

1.3 New Customers Testing

New customers who have not yet received their live Merchant ID can still perform an integration for testing purposes. Simply enter one of the below Test Merchant IDs and the use the Cardstream test cards to run a test transaction.

Standard Visa and MasterCard Testing use **100001**
3D Secure Testing use **100856**

This guide provides the information required to integrate with Cardstream, and gives a very basic example of code for doing so (further examples can be found on our website at www.cardstream.com). 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 Cardstream integration support.

If you do require programming assistance, please contact Cardstream on 0845 00 99 575 or via email to solutions@cardstream.com.

1.4 Pre-Requisites

You will need the following information to integrate with Cardstream direct.

Cardstream 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 details supplied to access the Merchant Management System. You should have received these details when your account was set up. You may also use test account IDs (listed above) and swap these for your live account details when you receive them.
Integration URL	https://gateway.cardstream.com/direct/

1.5 3D Secure

If your Merchant account is enrolled with 3D Secure, the Direct HTTP integration method will automatically attempt to perform 3D Secure transactions. If the customer's card does not participate in 3D Secure then the transaction will be processed as normal, otherwise the response may indicate that the issuing banks 3D Secure Access Control Server (ACS) needs to be contacted in order to authenticate the card holder.

When the 3D Secure authentication is required the Direct HTTP integration method 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 Merchant which must contain the URL of a page on the Merchant's server to return to when authentication has been completed.

On completion of the 3D Secure authentication the ACS will post the original **MD** along with a **PaRes** value to the **TermUrl**. These values should then be sent to CardStream in the **`threeDSMD`** and **`threeDSPaRes`** fields of a new request. This new request will complete the original transaction and return the normal response fields including any relevant 3D Secure authentication results.

Note: It is only necessary to send the **`threeDSMD`** and **`threeDSPaRes`** in the second request, however the Merchant 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)**.

You can choose how to deal with 3D Secure transactions that fail authentication – either declining the transaction or continuing without 3D Secure protection. These preferences are set in the Merchant Management System (MMS).

1.6 Test cards

For the latest copy of the test cards, for both 3D Secure and non 3D Secure transactions, please see Appendix A-9 & A10 below.

2 Gateway Request

The Merchant will need to send the request details to the integration URL via an HTTP POST request. The details should be URL encoded Name=Value fields separated by '&' characters (refer to RFC 1738 and the application/x-www-form-urlencoded media type).

For example, you might collect the customer information and card details on your own website and then send these via a direct socket connection to the CardStream server.

Please note that the field names are cAsE sEnSiTiVe.

2.1 General Fields

Field Name	Mandatory?	Description
merchantID	Yes	Your Cardstream Merchant user ID, or "100001" if you are just testing.
merchantPwd	No	The password you have configured for the merchantID. This is set within the MMS
signature	Yes	The hash used to sign the transaction request.
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.**
type	Yes	The type of transaction. Possible values are: **1** - Cardholder Not Present: Ecommerce. **2** - Cardholder Not Present: Mail Order. **3** - Point of Sale: Card Keyed. **4** - Point of Sale: Card Swiped. **5** - Point of Sale: Card Chip & Pin.
action	Yes	The transaction action. Possible values are: **PREAUTH**

		This will reserve an amount from the customer's card but not collect them. For a period of up to 5 days (depending on the card issuing bank) after the transaction is placed, you can place a subsequent transaction with an action of SALE and the XREF value returned from the first transaction in order to collect the previously reserved funds. If the period of time between the first and second transactions is greater than the card issuing bank reserves the funds for, then new, unreserved funds will be taken from the cardholders account. **SALE** This will collect an amount from the customer's card. **REFUND** This will refund an amount to the customer's card.
countryCode	Yes	ISO standard country code for the Merchant's location.
currencyCode	Yes	ISO standard 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.

2.2 Card Fields

Field Name	Mandatory?	Description
xref	No	If this field is set this transaction will be placed using the card data provided in the original transaction – Additionally, all other mandatory fields within section 2.2 are no longer required.

cardNumber	Yes	The customer's card number. **Numeric values only – no spaces or dashes**
cardCVV	Yes	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**
cardStartMonth	No	The customer's card start month. This is a two digit numeric printed on the front of the card. This field should be numeric, two digits in length and should be zero padded. For example; if the start month was January this would be sent as `01`, if the start month was November this would be sent as `11`. **Numeric, two characters only**
cardStartYear	No	The customer's card start year. This is a two digit numeric printed on the front of the card. **Numeric, two characters only**
cardExpiryMonth	Yes	The customer's card expiry month. This is a two digit numeric printed on the front of the card. This field should be numeric, two digits in length and should be zero padded. For example; if the expiry month was January this would be sent as `01`, if the expiry month was November this would be sent as `11`. **Numeric, two characters only**
cardExpiryYear	Yes	The customer's card expiry year. This is a two digit numeric printed on the front of the card. **Numeric, two characters only**
cardIssueNumber	No	The customer's card issue number. This is a numeric value printed on the front of the card. This field should only be provided if the card has an issue number (for example most Maestro cards)

		Numeric values only

2.3 Verification Field

The direct HTTP request, after completion, can POST the same transaction result data to a Callback URL in the background.

Field Name	Mandatory?	Description
`callbackURL`	No	A non-public URL which will receive a copy of the transaction result by POST.

2.4 Customer Details Fields

Customer details are optional by default, however if the Merchant has chosen to require AVS checking in their preferences, then **`customerAddress`** and **`customerPostCode`** become mandatory. All data is stored and accessible within the Merchant Management System (MMS).

Field Name	Mandatory?	Description
`customerName`	No	The customer or cardholder's name.
`customerAddress`	Yes, if AVS enabled	The customer or cardholder's address. For AVS checking this must be the registered billing address of the card.
`customerPostCode`	Yes, if AVS enabled	The customer or cardholder's post code. For AVS checking this must be the registered billing post code of the card.
`customerEmail`	No	The customer's email address.
`customerPhone`	No	The customer's telephone number.

2.5 American Express and Diners Card Fields

American Express or Diners Card cards require additional information about the customer's purchase to be included in the request. At least one order line must be included. For other cards types all items are optional and will be stored for reference purposed only and accessible within the Merchant Management System (MMS).

Field Name	Mandatory?	Description
item1Description	Yes†	A short text description of the item.
item1Quantity	Yes†	The quantity of the item purchased.
item1GrossValue	Yes†	The gross, or tax inclusive, value of this order line.
item2Description	No	A short text description of the item.
item2Quantity	No	The quantity of the item purchased.
item2GrossValue	No	The gross, or tax inclusive, value of this order line.
item3Description	No	A short text description of the item.
item3Quantity	No	The quantity of the item purchased.
item3GrossValue	No	The gross, or tax inclusive, value of this order line.
item4Description	No	A short text description of the item.
item4Quantity	No	The quantity of the item purchased.
item4GrossValue	No	The gross, or tax inclusive, value of this order line.
item5Description	No	A short text description of the item.
item5Quantity	No	The quantity of the item purchased.
item5GrossValue	No	The gross, or tax inclusive, value of this order line.

†These fields are only mandatory if an American Express or Diners Card is specified in the 'cardNumber' field.

With American Express or Diners Cards you may also provide tax ***or*** discount information. Once again for other cards types any values provided will be stored for reference purposes only and accessible within the Merchant Management System (MMS).

Field Name	Mandatory?	Description
taxValue	No	The total amount of tax for this order.
taxDiscountDescription	No	A text field to describe the tax applied *(e.g. "VAT at 20%")*

OR

Field Name	Mandatory?	Description
discountValue	No	The total amount of discount applied to this order.
taxDiscountDescription	No	A text field to describe the discount applied.

2.6 Merchant Data Field

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 CardStream†.

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**.

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

†*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.*

2.7 3D Secure Fields

After any 3D Secure authentication has been done by the card issuer's Access Control Server (ACS) the Merchant can repeat the request including the **`threeDSMD`** received in the original response and the **`PaRes`** information received from the ACS. These two fields can be passed alone in the request or along side other standard request fields. The **`threeDSMD`** marks the request as being a continuation and contains the necessary information to identify the initial request.

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

If the Merchant uses a separate third-party 3D Secure Merchant Plugin Interface (MPI) to authenticate with the card issuers Access Control Server then the 3D Secure authentication information provided from that MPI may be sent in a standard direct request. In this case the CardStream platform will use the supplied 3D Secure credentials and will not attempt to return a **`responseCode`** of **65802 (3DS AUTHENTICATION REQUIRED)**.

Field Name	Mandatory?	Description
`threeDSECI`	Yes	The Electronic Commerce Indicator (ECI) value returned from the 3rd party MPI.
`threeDSCAVV`	Yes	The Cardholder Authentication Verification Value (CAVV) returned from the 3rd party MPI.
`threeDSCAVVAlgorithm`	Yes	The CAVV algorithm used as returned from the 3rd party MPI.
`threeDSXID`	Yes	The unique identifier for the transaction as returned from the 3rd party MPI.

3 Gateway Response

The Cardstream Direct integration method returns data directly in response to the sent request. The data will also be sent to the Callback URL, if supplied, via an HTTP POST request. The data is returned as URL encoded Name=Value fields separated by '&' characters (refer to RFC 1738 and the application/x-www-form-urlencoded media type).

The fields initially sent to the integration URL are returned and in addition the following fields may be returned;

Please note that the field names are cAsE sEnSiTiVe.

3.1 General 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.
signature	Always	The hash used to sign the transaction reply.
xref	Always	The Merchant may store the cross reference for repeat transactions and refunds.
transactionUnique	If supplied	The value supplied in the initial request, if any.
amountReceived	On success	The amount of the transaction. This field used

		in conjunction with **transactionUnique** can help provide a measure of security.
transactionID	Always	The ID of the transaction on the Cardstream system – can be used to easily reconcile transactions in the administration panel.
orderRef	If supplied	The value supplied in the initial request, if any.
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.
cv2Check	Optional	Textual description of the AVS/CV2 CV2 check as described in Appendix A-4. Possible values are: **'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'**
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'**
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'**
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.

3.2 3D Secure Fields

When a 3D Secure transaction is required and no 3D Secure information has been provided then the following fields will be returned along with a **responseCode** of **65802 (3DS AUTHENTICATION REQUIRED)**.

For more information on how to process this response please refer to section 1.5.

Field Name	Returned?	Description
threeDSMD	Yes	A unique identifier required to continue the transaction after 3D Secure authentication by the issuers Access Control Server (ACS).
threeDSPaReq	Yes	The Payer Authentication Request to send to the Access Control Server (ACS).
threeDSACSURL	Yes	The URL of the the issuers Acess Control Server (ACS) to which the above PaReq must be sent.

The **threeDSMD** field is required to identify the transaction in order to complete it - this value must be stored by the Merchant while the 3D Secure authentication is being performed by the Access Control Server. If the Merchant would rather not store it locally it can be sent to the Access Control Server in the MD field which will be echoed back unchanged when 3D Secure authentication is completed.

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

Field Name	Returned?	Description
threeDSEnabled	Yes	The 3D Secure status of the Merchant account. Possible values are: **N** – the Merchant is not 3DS enabled **Y** – the Merchant is 3DS enabled
threeDSEnrolled	Yes	The 3D 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 3.2A-4 for further information.
threeDSAuthenticated	No	The 3D 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 3.2A-4 for further information.
threeDSPaReq	No	Payer Authentication Request (PaReq) that is sent to the Access Control Server (ACS) in order to verify the 3D Secure status of the credit card.
threeDSPaRes	No	Payer Authentication Response (PaRes) that is returned from the Access Control Server (ACS) determining the 3D Secure status of the credit card.
threeDSACSURL	No	The URL of the Access Control Server (ACS) to which the Payer Authentication Request (PaReq) should be sent.

threeDSECI	No	This contains a two digit Electronic Commerce Indicator (ECI) value, which is to be submitted in a credit card authorization message. This value indicates to the processor that the customer data in the authorization message has been authenticated. The data contained within this property is only valid if the **threeDSAuthenticated** value is **Y** or **A**.
threeDSCAVV	No	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	No	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	No	A unique identifier for the transaction as used in the 3D Secure process.
threeDSErrorCode	No	Any error response code returned by the 3D Secure Access Control Server (ACS) should there be an error in determining the card's 3D Secure status.
threeDSErrorDescription	No	Any error response description returned by the 3D Secure Access Control Server (ACS) should there be an error in determining the card's 3D Secure status.

threeDSMerchantPref	No	Any Merchant 3D Secure preference used to block or allow this transaction should the card not be authorized. These preferences can be set in the Merchant Management System (MMS).
threeDSVETimestamp	No	The time the card was checked for 3D Secure enrolment.
threeDSCATimestamp	No	The time the card was checked for 3D Secure authentication.

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. Refer to CardStream if this error occurs
64437	Reserved for future use. Refer to CardStream if this error occurs
64438	Reserved for future use. Refer to CardStream if this error occurs
64439	Invalid Credentials: **`merchantID`** is unknown
64440	Permission denied: caused by sending a request from an unauthorized IP address
64441	Reserved for future use. Refer to CardStream if this error occurs

64442	Request Mismatch: fields sent while completing a request do not match initially requested values. Usually due to sending different card details when completing a 3D Secure transaction to those used to authorise the transaction
64443	Request Ambiguous: request could be misinterpreted due to inclusion of mutually exclusive fields
64444	Request Malformed: couldn't parse the request data
64445	Suspended Merchant account
64446	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

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

65801	Reserved for future use. Refer to CardStream if this error occurs
65802	3D Secure authentication is required for this card
65803	3D Secure enrolment or authentication failure and Merchant 3DS 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. Refer to CardStream if this error occurs
66051	Reserved for internal use. Refer to CardStream if this error occurs
66052	Reserved for internal use. Refer to CardStream if this error occurs
66053	Reserved for internal use. Refer to CardStream if this error occurs
66054	Reserved for internal use. Refer to CardStream 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	Reserved for future use
66088	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

Invalid Request Field Error Codes: 66304 - 66559	
Code	**Description**
66304	Invalid request
66305	Invalid **`merchantID`** field
66306	Reserved for future use. Refer to CardStream if this error occurs
66307	Reserved for internal use. Refer to CardStream if this error occurs
66308	Reserved for internal use. Refer to CardStream if this error occurs
66309	Reserved for internal use. Refer to CardStream if this error occurs
66310	Reserved for internal use. Refer to CardStream 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 (reserved for future use)
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	Reserved for future use
66344	Reserved for future use
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
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
66418	Invalid item count. Tried to supply more than 6 line item details
66419	Invalid item sequence. Out of sequence line item details

A-2 Types of card

The following is a list of card types which may be returned by the gateway.

Card Code	Card Type
AM	American Express
CF	Clydesdale Financial Services
DI	Diners Club
EL	Electron
JC	JCB
MA	International Maestro
MC	Mastercard
SO	Solo
ST	Style
SW	Domestic Maestro (Formerly Switch)
VC	Visa Credit
VD	Visa Debt
VP	Visa Purchasing

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 3D Secure Enrolment/Authentication Codes

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

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

N - Not Enrolled: The checked card is eligible for the 3DSecure (it is within the card association's range of accepted cards) but the card issuing bank does not participate in the 3D 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 3D 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 CardStream system encountered an error. This card is flagged as 3D 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 3D 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 CardStream system 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-5 3D Secure Enrolment/Authentication Only

Normally the direct HTTP interface will perform most of the 3D 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 the Merchant may wish to gain more control over the Enrolment and Authentication process. The following field allows the request processing to stop after the 3D Secure enrolment check or authentication check and return;

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

As this stop is requested by the Merchant then **`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 the Merchant to continue to the next 3D Secure stage when required. In order to continue the process the **`threeDSMD`** field is returned along with any relevant 3D Secure response fields suitable for that stage in the processing.

If this flag is used when 3D Secure is not enabled on the account or after the 3D 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 3D Secure enrolment or authentication only.

A-6 Request Checking Only

Sometimes the Merchant may wish to submit a request via the Direct 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 CardStream and not available in the Merchants Management System (MMS).*

A-7 3D Secure Example Code

The following example shows how to send a 3D Secure request to the direct HTTP method and decode the response;

/myshop/processtransaction.php:

```
<?PHP

// PreShared Key entered on MMS. The demo accounts is fixed, but merchant accounts can be
updated from the MMS.
$pre_shared_key = "Circle4Take40Idea";

// Merchant Password
$merchant_password = "Carry21After46Pardon";

// Hasing Method, Supported Methods are: SHA512 (preferred), SHA256, SHA1, MD5, CRC32
$hashing_method = "SHA512";

// Build Request
$req = array(
    "merchantID" => "100856",
    "action" => "SALE",
    "type" => 1,
    "transactionUnique" => uniqid(),
    "currencyCode" => 826,
    "amount" => 1001,
    "orderRef" => "Test purchase",
    "cardNumber" => "4012001037141112",
    "cardExpiryMonth" => 12,
    "cardExpiryYear" => 15,
    "cardCVV" => '083',
    "customerName" => "CardStream",
    "customerEmail" => "solutions@cardstream.com",
    "customerPhone" => "+44 (0) 845 00 99 575",
    "customerAddress" => "16 Test Street",
    "countryCode" => 826,
    "customerPostCode" => "TE15 5ST",
    "merchantPwd" => $merchant_password,
    "threeDSMD" => (isset($_REQUEST['MD']) ? $_REQUEST['MD'] : null),
    "threeDSPaRes" => (isset($_REQUEST['PaRes']) ? $_REQUEST['PaRes'] : null),
    "threeDSPaReq" => (isset($_REQUEST['PaReq']) ? $_REQUEST['PaReq'] : null)
);

// Data must be sorted by key
ksort($req);

// Build the signature field and concatinate the key to the end
$signature_fields = http_build_query($req) . $pre_shared_key;

// Make a hash of the fields
$hash = hash($hashing_method, $signature_fields);

// Add Signature field to the end of the requst. If you are using the the default hasing
method (SHA512) it does not need to be sent
$req['signature'] = ( $hashing_method != "SHA512" ? "{" . $hashing_method . "}" : "" ) .
$hash;


$ch = curl_init('https://gateway.cardstream.com/direct/');
curl_setopt($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($ch, CURLOPT_RETURNTRANSFER, true);
parse_str(curl_exec($ch), $res);
curl_close($ch);

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>";
} elseif (isset($res['signature'])) {
        $return_signature = $res['signature'];

        // Remove the signature as this isn't hashed in the return
        unset($res['signature']);

        // Sort the returned array
        ksort($res);

        // The returned hash will always be SHA512
        if ($return_signature == hash("SHA512", http_build_query($res) . $pre_shared_key)) {
                echo "<p>Signature Check OK!</p>" . PHP_EOL;
                if ($res['responseCode'] === "0") {
                        echo "<p>Thank you for your payment</p>" . PHP_EOL;
                } else {
                        echo "<p>Failed to take payment: " .
htmlentities($res['responseMessage']) . "</p>" . PHP_EOL;
                }
        } else {
                die("Sorry, the signature check failed");
        }
} else {
        if ($res['responseCode'] === "0") {
                echo "<p>Thank you for your payment</p>";
        } else {
                echo "<p>Failed to take payment: " . htmlentities($res['responseMessage']) .
"</p>" . PHP_EOL;
        }
}
?>
```

A-8 Non 3D Secure Example Code

The following example shows how to send a non 3D Secure request to the direct HTTP method and decode the response:

/myshop/processtransaction.php:

```
<?PHP

// PreShared Key entered on MMS. The demo accounts is fixed, but merchant accounts can be
updated from the MMS.
$pre_shared_key = "Circle4Take40Idea";

// Merchant Password
$merchant_password = "Carry21After46Pardon";

// Hasing Method, Supported Methods are: SHA512 (preferred), SHA256, SHA1, MD5, CRC32
$hashing_method = "SHA512";

// Build Request
$req = array(
    "merchantID" => "100001",
    "action" => "SALE",
    "type" => 1,
    "transactionUnique" => uniqid(),
    "currencyCode" => 826,
    "amount" => 1001,
    "orderRef" => "Test purchase",
    "cardNumber" => "4929421234600821",
    "cardExpiryMonth" => 12,
    "cardExpiryYear" => 15,
    "cardCVV" => 356,
    "customerName" => "CardStream",
    "customerEmail" => "solutions@cardstream.com",
    "customerPhone" => "+44 (0) 845 00 99 575",
    "customerAddress" => "6347 Test Card Street",
    "countryCode" => 826,
    "customerPostCode" => "17TST8",
    "merchantPwd" => $merchant_password,
);

// Data must be sorted by key
ksort($req);

// Build the signature field and concatinate the key to the end
$signature_fields = http_build_query($req) . $pre_shared_key;

// Make a hash of the fields
$hash = hash($hashing_method, $signature_fields);

// Add Signature field to the end of the requst. If you are using the the default hasing
method (SHA512) it does not need to be sent
$req['signature'] = ( $hashing_method != "SHA512" ? "{" . $hashing_method . "}" : "" ) .
$hash;


$ch = curl_init('https://gateway.cardstream.com/direct/');
curl_setopt($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($ch, CURLOPT_RETURNTRANSFER, true);
parse_str(curl_exec($ch), $res);
curl_close($ch);

if (isset($res['signature'])) {
        $return_signature = $res['signature'];
```

```
        // Remove the signature as this isn't hashed in the return
        unset($res['signature']);

        // Sort the returned array
        ksort($res);

        // The returned hash will always be SHA512
        if ($return_signature == hash("SHA512", http_build_query($res) . $pre_shared_key)) {
                echo "<p>Signature Check OK!</p>" . PHP_EOL;
                if ($res['responseCode'] === "0") {
                        echo "<p>Thank you for your payment</p>" . PHP_EOL;
                } else {
                        echo "<p>Failed to take payment: " .
htmlentities($res['responseMessage']) . "</p>" . PHP_EOL;
                }
        } else {
                die("Sorry, the signature check failed");
        }
} else {
        if ( $res['responseCode'] === "0" ) {
                echo "<p>Thank you for your payment</p>";
        } else {
                echo "<p>Failed to take payment: " . htmlentities($res['responseMessage']) .
"</p>" . PHP_EOL;
        }
}
?>
```

A-9 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

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*

* If applicable to transaction / merchant / acquirer type

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-10 3D Secure Test Cards

3D Secure test cards for MasterCard using SecureCode

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

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

3D Secure test cards for Visa using Verified by Visa

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

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)

A-11 Signing Your Request

A message can be signed by hashing the whole URL encoded Name=Value request string with a secret passphrase appended. This security passphrase can be configured on a per merchant account basis in the Merchant Management System (MMS).

Care must be taken to normalise any embedded line ending to just use a single New Line character (ascii character 10).

Various hashing algorithms are supported allowing you to choose the one most suitable for your integration language. SHA512 is the default and preferred, if using an algorithm other than SHA512 then the algorithm name should be pre-pended to the hash enclosed in braces.

The following algorithms are supported (from most secure to least secure order): SHA512, SHA256, SHA1, MD5, CRC32.

The hash must be sent in the signature field. This field must not be included in the message that is used to generate the hash.

Note: when a secret is configured for the merchant account then every

message must be signed – failure to sign a message will cause it to be rejected due to a missing signature. The gateway will also sign any response and any details POSTed to any callback URL using the same signature allowing the merchant to verify that any response has not been tampered with.