Transparent Redirect

takepayments gateway

V1.1 – 16th December 2019

takepayments
we make it easy

takepayments
we make it easy

Contents

Introduction

Intended Audience

This document is technical in nature and should be used by your company's developers to integrate your systems into the payment gateway. It assumes that the reader has knowledge and understanding of basic HTML concepts such as form post.

Simplifying the Integration Process

There are many complexities when dealing with card transactions. If you try and tackle them all at once the task of integrating will seem complicated. The best way to do the integration is to follow a simple step by step approach and break the process down into manageable sections, each adding functionality as you go along.

To assist you example code is available in the resource section in most of the common programming languages. Where possible please use these well documented examples as a starting point.

Adhering to good coding practices will also greatly simplify your task.

takepayments
we make it easy

IMPORTANT INFORMATION: PLEASE READ CAREFULLY

Important Notes

Gateway URLs

The generalised full URL to use in your posts to the Hosted Payment Form is
https://mms.tponlinepayments2.com/Pages/PublicPages/TransparentRedirect.aspx

Hashing Explained

When the Hosted Payment Form solution is implemented, the transaction data must be protected as it is being passed to the payment page via the customer's browser. The data is protected by use of Hashing. Hashing is used to produce a unique "signature" for the data being passed (it is generated using not only the data being transmitted, but also secret data that is not transmitted, so it is impossible to recreate the hash digest with just the data that is passed via their browser). The hash signature is then re-calculated by our system on receipt of the transmitted data, and if it does not match the hash signature that was transmitted with the data, then the data has been tampered with, and the transaction will stop with an error message. The same process (in reverse) should be carried out by your site on receipt of the transaction results.

Examples of this type of tampering could be lowering the transaction price (say from £100.00 down to $1.00) or making a failed transaction look like an authorised one. This is called a "man-in-the-middle" attack.

Simple hashing example

Here is an example of some transaction variables:

```
MerchantID: YOURCO-1234567
Amount: 100.00
CurrencyCode: 826
OrderID: 12345
```

These variables would be concatenated (in a specific – see variables tables below) with data known only to your system and ours (the merchant password and pre shared key) which is NOT transmitted with the transaction request. This produces the following string:

```
MerchantID=YourCo-1234567&Password=MyPassword&PreSharedKey=ASecretKey&Amount=10000
&CurrencyCode=826&OrderID=12345
```

A simple hash method would output the following hash digest (or "Signature"), when this string is passed into a hashing (in this case SHA1) function (which is also transmitted with the transaction variables):

```
5c6b9c913b2301e9aa6ff488b06e09273cded2a5
```

If the amount was altered from £100.00 to £1.00:

```
MerchantID: YourCo-1234567
Amount: 1.00
CurrencyCode: 826
OrderID: 12345
```

When these variables are received by our system, they would be used to produce the following string:

```
MerchantID=YourCo-1234567&Password=MyPassword&PreSharedKey=ASecretKey&Amount=100&C
urrencyCode=826& OrderID=12345
```

Which when passed into the same hashing function would produce the following hash digest (or "Signature"):

```
4ba1164acbec732c18cd6e5f632adcdd4b440237
```

This demonstrates that changing any of these variables, even just a single character, results in a very different resulting hash digest, and makes the process of detecting variable.

Notation Explained

The communication between the merchant's system/customer's browser, to the gateway PaymentForm are POST via HTML form variables. The below table or similar format is used throughout this document to help explain the requirements for passing/receiving variables to/from the gateway.

Input Field Name	Data Type	Max Length	Mandatory or Always Present	Comments
InputName				

An example HTML form POST variable which would get to the gateway;

```
<input type="hidden" name="InputName" value="InputValue" />
```

Input Field Name:

The value in this field is case sensitive and should be sent exactly as is stated here. Failure to do so will result in errors, specifically relating to missing variables or hash digest mismatches.

Data Type:

All variables will be converted to a string when the HTML is rendered anyway, but this is to stipulate what the expected value should be readable as. For example, Boolean should only be sent as "TRUE" or "FALSE". Some scripting languages like PHP only state "1" or "0" for Boolean, which would be deemed invalid by the gateway so all values must be converted to the noted data type first.

Max Length:

This is the maximum length that the gateway will allow for the variable. If a "-" is noted, this means it is of variable length. If there is a numerical value in the field, any variables passed that are longer than this will result in an error.

Mandatory:

This informs the merchant if a variable is deemed to be mandatory by the gateway. Failure to send mandatory variables to the gateway will result in an error. If this is marked with "Yes", check the comments section also as there may be specific situations when it is not mandatory.

Comments:

This field should be read thoroughly to determine if they are relevant to the integration you are performing. This field may contain information explaining more in depth the variables use and/or assist in determining if it is indeed mandatory or not.

Data Types:

Data Type	Description
N	Numeric - only numbers allowed
A	Alpha - any printable character is allowed
DT	Date Time stamps
B	Boolean - only TRUE or FALSE are allowed - these are passed as strings for the hash and form (i.e. "true" instead of 1)
-	Special types - these variables only allow a specific set of values. Details of the allowed value are in the comments section.

Creating and hashing payment form

Payment Form Variables

Below is a description of the variables that comprise the input API of the payment form. These variables are delivered as form variable.

Please note that the string to hash will need to be created with the fields in the order of the table below. Optional fields can either be omitted from the string to hash and the form or sent as blank/null/empty.

When creating the string to hash, it is important to ensure that the same fields that are being sent in the form are included in the hash and that both fields match. Optional fields if included in the string to hash will need to be sent empty in the form to ensure that HashDigest matches.

Variable Name	Data Type	Max Length	Mandatory or Always Present	Comments
HashDigest	A	-	Yes	A hashed string that contains all the variables passed and data that is not passed but is known to both sides - namely the PreSharedKey and the gateway account password. (see section below)
MerchantID	A	15	Yes	The merchant ID that corresponds to the gateway account the transaction will be run through. NOTE: If this variable is not present, then the skinning of the payment form will not happen
Amount	N	13	Yes	The transaction amount in minor currency - e.g. for £10.00, it must be submitted as 1000
CurrencyCode	N	3	Yes	The currency of the transaction. ISO 4217 e.g. GBP: 826
EchoAVSCheckResult	B	true/ false	Yes	Instructs the payment form to include the AVS check result of the transaction in the output variables
EchoCV2CheckResult	B	true/ false	Yes	Instructs the payment form to include the CV2 check result of the transaction in the output variables
EchoThreeDSecureAuthenticationCheckResult	B	true/ false	Yes	Instructs the payment form to include the 3D Secure check result of the transaction in the output variables
EchoCardType	B	true/ false	Yes	Instructs the payment form to include the card type of the transaction in the output variables
AVSOverridePolicy	A	4	No	Sets the override AVS checking policy for this transaction
CV2OverridePolicy	A	2	No	Sets the CV2 checking policy for this transaction
ThreeDSecureOverridePolicy	B	true/ false	No	Instructs the payment form to enable/disable the 3D Secure checking for this transaction (where possible)
OrderID	A	50	Yes	A merchant side ID for the order - primarily used to for determining duplicate transactions. Note: make sure that special characters in the OrderID are properly

Variable Name	Data Type	Max Length	Mandatory or Always Present	Comments
				escaped, otherwise the hash digest will not match
TransactionType	-	-	Yes	Must be either SALE or PREAUTH
TransactionDateTime	DT	-	Yes	The date & time (as seen by the merchant's server) of the transaction. Needs to be in the form "YYYY-MM-DD HH:MM:SS +OO:OO", with the time in 24-hour format, where OO:OO is the offset from UTC - e.g. "2008-12-01 14:12:00 +01:00"
CallbackURL	A	-	Yes	The URL of the page on the merchant's site that the results of the transaction will be posted back to (see section below)
OrderDescription	A	256	No	A description for the order. Note: make sure that special characters in the OrderDescription are properly escaped, otherwise the hash digest will not match
Address1	A	100	No	Customer's billing address line 1
Address2	A	50	No	Customer's billing address line 2
Address3	A	50	No	Customer's billing address line 3
Address4	A	50	No	Customer's billing address line 4
City	A	50	No	Customer's billing address city
State	A	50	No	Customer's billing address state
PostCode	A	50	No	Customer's billing address post code
CountryCode	N	3	No	Customer's billing country code. ISO 3166-1 e.g. United Kingdom: 826
EmailAddress	A	100	No	Customer's email address
PhoneNumber	A	30	No	Customer's phone number
CardName	A	100	Yes	The cardholder's name as it appears on the front of the card
CardNumber	A	19	Yes	The full card number as it appears on the front of the card
ExpiryDateMonth	A	2	Yes	The first 2 digits from the card's expiry date representing the month part of the expiry date e.g. 02/12 where 02 is the card's expiry month
ExpiryDateYear	A	2	Yes	The last 2 digits from the card's expiry date representing the year part of the expiry date e.g. 02/12 where 12 is the card's expiry year of 2012
StartDateMonth	A	2	No	The first 2 digits from the card's start date representing the month part of the start date e.g. 01/09 where 01 is the card's start month
StartDateYear	A	2	No	The last 2 digits from card's start date representing the year part of the start date e.g. 01/09 where 09 is the card's start year of 2009
IssueNumber	A	2	No	The card's issue number as it appears on the front on the card
CV2	N	3	No	The card's CV2 number

Incoming Hash Digest Variables

Below is the order that the variables should be listed when creating the hash digest. The string to be hashed must be comprised of the variables listed in the order below in standard URL format (i.e. listed in name/value pairs, delimited with an ampersand character, for example:

```
"variable1=value&variable2=value&variable3=value"
```

The variable names and values are case-sensitive, and the values should be represented EXACTLY as they appear in the form (NON-URL ENCODED). This hash must be checked against the one submitted in the form, and it should be the same as the hash digest created by us. Any differences should be treated with EXTREME caution, as this indicates that the variables in the form have been tampered with.

Variable Name	Mandatory or Always Present	Comments
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
MerchantID	Yes	
Password	Yes	
Amount	Yes	
CurrencyCode	Yes	
OrderID	Yes	Note: make sure that special characters in the OrderID are properly escaped, otherwise the hash digest will not match
TransactionType	Yes	
TransactionDateTime	Yes	
CallbackURL	Yes	
OrderDescription	Yes	Must be included as "OrderDescription=" if not submitted or empty in the form. Note: make sure that special characters in the OrderDescription are properly escaped, otherwise the hash digest will not match

3D Secure "Authentication Required"

Output Variables

Below is a description of the variables that will be posted to the merchant's CallbackURL when the cardholder needs to authenticate themselves directly with their bank. This will be returned in response to the **initial** payment request.

Variable Name	Data Type	Max Length	Comments
HashDigest	A	-	A hashed string that contains all the variables passed and data that is not passed but is known to both sides - namely the PreSharedKey and the gateway account password. (see section below)
MerchantID	A	15	The merchant ID that was used to process the transaction
StatusCode	N	-	This indicates the status of the transaction
Message	A	-	This is the message returned by the gateway
CrossReference	A	25	The cross reference of the transaction returned by the gateway
OrderID	A	-	The OrderID representing the transaction. Note: make sure that special characters in the OrderID are properly escaped, otherwise the hash digest will not match
TransactionDateTime	DT	-	The date & time (as seen by the gateway server) of the transaction. Will be in the form "YYYY-MM-DD HH:MM:SS +OO:OO", with the time in 24-hour format, where OO:OO is the offset from UTC - e.g. "2008-12-01 14:12:00 +01:00"
ACSURL	A	-	If the card has been determined as requiring 3D Secure authentication, this gives the URL of the ACS server that the PaREQ must be sent to
PaREQ	A	-	If the card has been determined as requiring 3D Secure authentication, this gives the base64 encoded payment request that must be passed to the ACS for authentication. This must be sent to the ACS as "PaReq"

Outgoing Hash Digest

Below is the order that the variables will be listed when creating the 3D Secure "authentication required" payment response hash digest to check against the one in submitted in the form. The string to be hashed must be comprised of the variables listed in the order below in standard URL format (i.e. listed in name/value pairs, delimited with an ampersand character for example,

```
"variable1=value&variable2=value&variable3=value"
```

The variable names and values are case-sensitive, and the values should be represented EXACTLY as they appear in the form (NON-URL ENCODED). This hash must be checked against the one submitted in the form, and it should be the same as the hash digest created by us. Any differences should be treated with EXTREME caution, as this indicates that the variables in the form have been tampered with.

Variable Name	Mandatory or Always Present	Comments
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
MerchantID	Yes	
Password	Yes	
StatusCode	Yes	
Message	Yes	
CrossReference	Yes	
OrderID	Yes	Note: make sure that special characters in the OrderID are properly escaped, otherwise the hash digest will not match
TransactionDateTime	Yes	
ACSURL	Yes	
PaREQ	Yes	

3D Secure "Post Authentication"

Input Variables

Below is a description of the variables that comprise the input API of the 3D Secure Authentication portion of the payment form. This request will be sent after the cardholder has authenticated themselves directly with their bank and will yield a "Payment Complete" output response.

Variable Name	Data Type	Max Length	Comments
HashDigest	A	-	A hashed string that contains all the variables passed and data that is not passed but is known to both sides - namely the PreSharedKey and the gateway account password. (see section below)
MerchantID	A	15	The merchant ID that was used to process the transaction
CrossReference	A	25	This is the unique cross reference for this transaction. If the transaction was determined to be a duplicate transaction, this value will hold the cross reference of the previous transaction, which this transaction was deemed a duplicate of
TransactionDateTime	DT	-	The date & time (as seen by the merchant's server) of the transaction. Needs to be in the form "YYYY-MM-DD HH:MM:SS +OO:OO", with the time in 24-hour format, where OO:OO is the offset from UTC - e.g. "2008-12-01 14:12:00 +01:00"
CallbackURL	A	-	The URL of the page on the merchant's site that the results of the transaction will be posted back to (see section below)
PaRES	A	-	The base64 encoded payment response (PaRES) string returned by the interaction with the ACS server

Incoming Hash Digest

Below is the order that the variables should be listed when creating the 3D Secure authentication request hash digest to check against the one in submitted in the form. The string to be hashed must be comprised of the variables listed in the order below in standard URL format (i.e. listed in name/value pairs, delimited with an ampersand character for example,

```
"variable1=value&variable2=value&variable3=value"
```

The variable names and values are case-sensitive, and the values should be represented EXACTLY as they appear in the form (NON-URL ENCODED). This hash must be checked against the one submitted in the form, and it should be the same as the hash digest created by us. Any differences should be treated with EXTREME caution, as this indicates that the variables in the form have been tampered with.

Variable Name	Mandatory or Always Present	Comments
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
MerchantID	Yes	
Password	Yes	
CrossReference	Yes	
TransactionDateTime	Yes	
CallbackURL	Yes	
PaRES	Yes	

Payment Complete

Output Variables

Below is a description of the variables will be posted to the merchant's CallbackURL. These comprise the "Payment Complete" output API of the payment form.

Variable Name	Data Type	Max Length	Comments
HashDigest	A	-	A hashed string that contains all the variables passed and data that is not passed but is known to both sides - namely the PreSharedKey and the gateway account password. (see section below)
MerchantID	A	15	The merchant ID that was used to process the transaction
StatusCode	N	-	This indicates the status of the transaction: • 0: transaction successful • 5: card referred • 5: card declined • 20: duplicate transaction • 30: exception
Message	A	512	This gives a more detailed description of the status of the transaction
PreviousStatusCode	N	-	If the transaction was deemed to be a duplicate transaction, this indicates the status of the previous transaction
PreviousMessage	A	512	If the transaction was deemed to be a duplicate transaction, this gives a more detailed description of the status of the previous transaction
CrossReference	A	25	This is the unique cross reference for this transaction. If the transaction was determined to be a duplicate transaction, this value will hold the cross reference of the previous transaction, which this transaction was deemed a duplicate of
AddressNumericCheckResult	A	-	If requested (input variable "EchoAVSCheckResult = true") this gives the results of the address numeric check - will be PASSED, FAILED, PARTIAL, NOT_CHECKED or UNKNOWN
PostCodeCheckResult	A	-	If requested (input variable "EchoAVSCheckResult = true") this gives the results of the post code check - will be PASSED, FAILED, PARTIAL, NOT_CHECKED or UNKNOWN
CV2CheckResult	A	-	If requested (input variable "EchoCV2CheckResult = true") this gives the results of the CV2check - will be PASSED, FAILED, NOT_CHECKED or UNKNOWN
ThreeDSecureAuthenticationCheckResult	A	-	If 3D Secure policy is enabled (input variable "ThreeDSecureOverridePolicy = true") this will give the results of the 3D Secure check
CardType	A	-	If requested (input variable "EchoCardType = true") this gives the card type of the transaction
CardClass	A	-	If requested (input variable "EchoCardType = true") this gives the card class of the transaction
CardIssuer	A	-	If requested (input variable "EchoCardType = true") this gives the card issuer (if known)
CardIssuerCountryCode	N	3	If requested (input variable "EchoCardType = true") this gives the 3-digit code of the country the card was issued in (if known)
Amount	N	13	The amount, in minor currency, of the transaction that was processed

Variable Name	Data Type	Max Length	Comments
CurrencyCode	N	3	The currency code of the transaction that was processed. ISO 4217 e.g. GBP: 826
OrderID	A	50	The order ID of the transaction that was processed. Note: make sure that special characters in the OrderID are properly escaped, otherwise the hash digest will not match
TransactionType	-	-	The transaction type of the transaction that was processed. Will be either SALE or PREAUTH
TransactionDateTime	DT	-	The date & time (as seen by the gateway server) of the transaction. Will be in the form "YYYY-MM-DD HH:MM:SS +OO:OO", with the time in 24-hour format, where OO:OO is the offset from UTC - e.g. "2008-12-01 14:12:00 +01:00"
OrderDescription	A	256	The order description of the transaction that was processed. Note: make sure that special characters in the OrderDescription are properly escaped, otherwise the hash digest will not match
Address1	A	100	Customer's billing address line 1 as it was submitted to the gateway
Address2	A	50	Customer's billing address line 2 as it was submitted to the gateway
Address3	A	50	Customer's billing address line 3 as it was submitted to the gateway
Address4	A	50	Customer's billing address line 4 as it was submitted to the gateway
City	A	50	Customer's billing city as it was submitted to the gateway
State	A	50	Customer's billing state as it was submitted to the gateway
PostCode	A	50	Customer's billing post code as it was submitted to the gateway
CountryCode	N	3	Customer's billing country code as it was submitted to the gateway. ISO 3166-1 e.g. United Kingdom: 826
EmailAddress	A	100	The customer's email address as it was submitted to the gateway
PhoneNumber	A	30	The customer's phone number as it was submitted to the gateway

takepayments
we make it easy

Outgoing Hash Digest

Below is the order that the variables will be listed when creating the hash digest to check against the one in submitted in the form. The string to be hashed must be comprised of the variables listed in the order below in standard URL format (i.e. listed in name/value pairs, delimited with an ampersand for example,

```
"variable1=value&variable2=value&variable3=value"
```

The variable names and values are case-sensitive, and the values should be represented EXACTLY as they appear in the form (NON-URL ENCODED). This hash must be checked against the one submitted in the form, and it should be the same as the hash digest created by us. Any differences should be treated with EXTREME caution, as this indicates that the variables in the form have been tampered with.

Variable Name	Mandatory or Always Present	Comments
PreSharedKey	See comments	The pre shared key should ONLY be included in the hash digest if the chosen hash method is standard (i.e. not HMAC) MD5 or SHA1. If the chosen hash method is either HMACMD5 or HMACSHA1, then the pre shared key is used as part of the hash generation so should be ENTIRELY omitted from the string to be hashed - if it is present in these cases (even as an empty string), then an error will be thrown
MerchantID	Yes	
Password	Yes	
StatusCode	Yes	
Message	Yes	
PreviousStatusCode	Yes	Must be included as "PreviousStatusCode=" if an empty variable in the form
PreviousMessage	Yes	Must be included as "PreviousMessage=" if an empty variable in the form
CrossReference	Yes	
AddressNumericCheckResult	Yes	Must be included as "AddressNumericCheckResult=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
PostCodeCheckResult	Yes	Must be included as "PostCodeCheckResult=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
CV2CheckResult	Yes	Must be included as "CV2CheckResult=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
ThreeDSecureAuthenticationCheckResult	Yes	Must be included as "ThreeDSecureAuthenticationCheckResult=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
CardType	Yes	Must be included as "CardType=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
CardClass	Yes	Must be included as "CardClass=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)

Variable Name	Mandatory or Always Present	Comments
CardIssuer	Yes	Must be included as "CardIssuer=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
CardIssuerCountryCode	Yes	Must be included as "CardIssuerCountryCode=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
Amount	Yes	
CurrencyCode	Yes	
OrderID	Yes	Note: make sure that special characters in the OrderID are properly escaped, otherwise the hash digest will not match
TransactionType	Yes	
TransactionDateTime	Yes	
OrderDescription	Yes	Must be included as "OrderDescription=" if an empty variable in the form. Note: make sure that special characters in the OrderDescription are properly escaped, otherwise the hash digest will not match
Address1	Yes	Must be included as "Address1=" if an empty variable in the form
Address2	Yes	Must be included as "Address2=" if an empty variable in the form
Address3	Yes	Must be included as "Address3=" if an empty variable in the form
Address4	Yes	Must be included as "Address4=" if an empty variable in the form
City	Yes	Must be included as "City=" if an empty variable in the form
State	Yes	Must be included as "State=" if an empty variable in the form
PostCode	Yes	Must be included as "PostCode=" if an empty variable in the form
CountryCode	Yes	Must be included as "CountryCode=" if an empty variable in the form
EmailAddress	Yes	Must be included as "EmailAddress=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)
PhoneNumber	Yes	Must be included as "PhoneNumber=" if an empty variable in the form. NOT included in the hash if not present in the form (only for backwards compatibility)

Appendix

Appendix 1: Gateway Response StatusCodes

Below are the status codes likely to be received when integrating with the gateway.

Status Code	Transaction Result	Description
0	Successful	**Transaction Authorised:** The transaction was successful, and you will be given an Authorisation Code as part of the message returned by the gateway.
3	Incomplete	**Transaction Awaiting 3D Secure Authentication:** Transaction is now awaiting 3D Secure Authentication from the customer. This status has a 2-hour expiry time set by the card scheme, at which point, the transaction will fail (Issuer Authentication Expired).
4	Referred	**Transaction Referred:** The card issuer has parked the transaction awaiting contact with the customer before proceeding to authorise or decline the transaction.
5	Declined	**Transaction Failed:** The transaction was declined by the card issuer or acquiring bank. In the event of the Address or CV2 verification failure, this will also be noted on the message from the gateway (Example, "Card declined: AVS policy + CV2 policy"). If the message given by the gateway only says "Card declined" with no other information, then no other information was given to us from the card issuer or acquiring bank as to the underlying reason why. The only person who can find out why the transaction was declined is the customer by contacting their bank directly.
20	Duplicate Transaction	The transaction which was processed was a duplicate. If this is the case, then the original transaction information is also passed back from the gateway so you can determine the result of the original transaction. Please refer to your respective integration method documentation form more information.
30	Failed (Error(s) Occurred)	**Transaction Failed:** This is usually an indicator that the integration to the gateway is incomplete and/or not working correctly. There will also be additional error information feedback from the gateway for merchants to determine what the error is specifically. Please refer to your respective integration methods documentation for more information.

Appendix 2: Transaction Data Flow

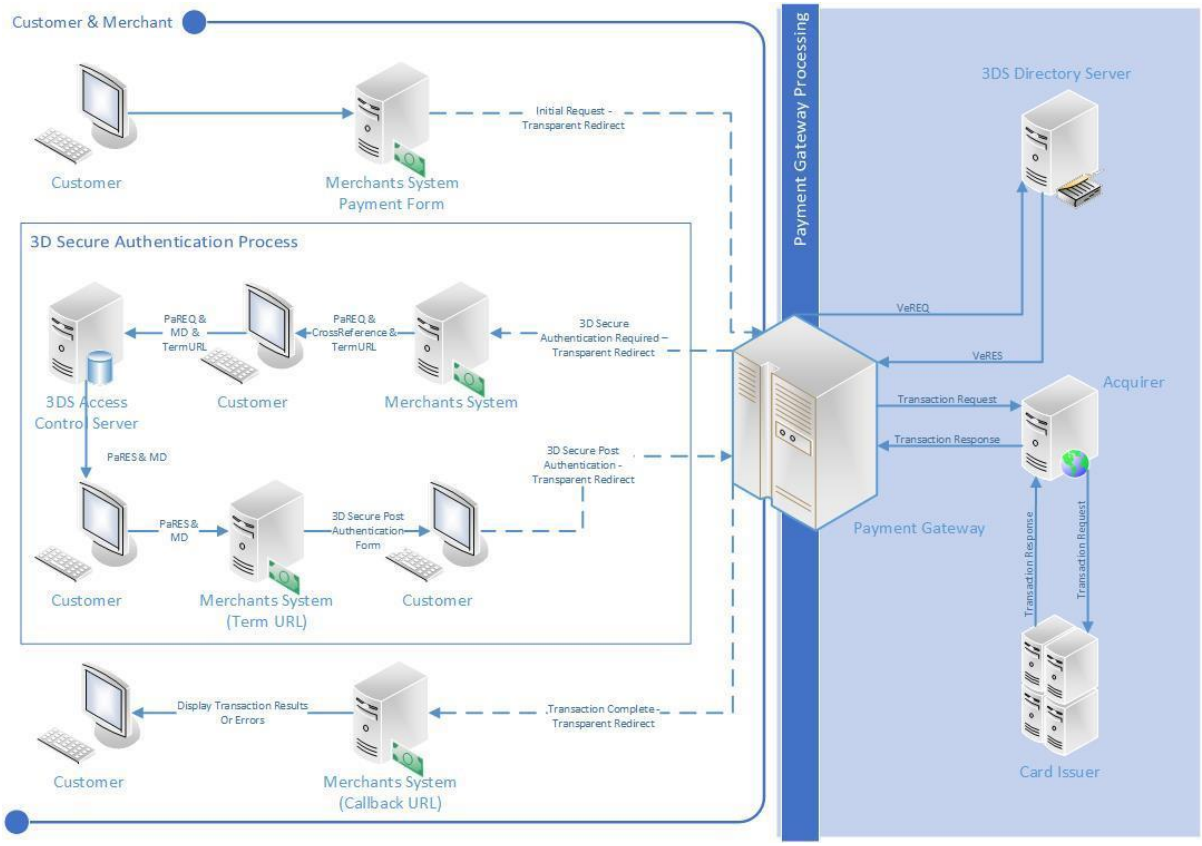
Listed below are the steps that a Transparent Redirect transaction will take. There are also 3 diagrams to show the transaction data flow in different scenarios.

1. The cardholder navigates to the merchant's website and supplies their card details into the merchant's payment form. The payment form is hosted directly on the merchant's system.
2. The Merchant and transactional data, optionally along with Customer information are passed to the payment gateway (Transparent Redirect URL), as part of a transparent redirect. The customer is unaware of this redirection as nothing changes on screen whilst processing takes place. The data passed to the payment gateway will be checked for errors at this point.
    a. If errors occur (for example; Variable Tampering), the payment gateway doesn't allow the transaction to go any further and the error details are passed back to the Merchant's system (CallbackURL) and moves to step 11.
    b. If no errors occurred, the transaction moves to step 3.
3. The payment gateway contacts the Directory Server to query whether this card is enrolled in the 3D Secure scheme.
4. The Directory Server determines whether the card is enrolled in the 3DS scheme, then passes this information back to the payment gateway.
    a. If the card is enrolled in the 3D Secure Authentication Scheme, the transaction moves to step 5.
    b. If not, the transaction moves to step 10.
5. The payment gateway passes the URL of the cardholder's bank's Access Control Server (ACSURL) and additional data from which a Payment Request string (PaREQ) is created, to the merchant's system (CallbackURL) as part of a transparent redirect. Again, the customer is unaware of this redirect. The data passed to the Merchant's System should be checked for errors at this point.
    a. If errors occur (for example; Variable Tampering), the transaction shouldn't go any further and moves to step 11.
    b. If no errors occurred, the transaction moves to step 6.
6. The customer is then redirected by the merchant's system (CallbackURL) to their bank's Access Control Server (ACSURL) and they are greeted with the last 4 digits of their credit card & the identification text they specified when registering their card for 3D Secure. This redirection is not transparent; it is very much visible to the customer.
7. The customer then validates their card details using their 3D Secure password, which is validated by their bank's Access Control Server.
8. The Access Control Server then initiates a redirect of the customer's browser back to a secure processing page on the merchant's website (TermURL), which forwards the payment response string (PaRES) from the Access Control Server to the payment gateway (Transparent Redirect URL) using a transparent page redirect. The data passed to the payment gateway will be checked for errors at this point.
    a. If errors occur (for example; Variable Tampering), the details will be passed back to the merchant's system (CallbackURL) and the transaction won't go any further.
    b. If no errors occurred, the transaction moves to step 10.
9. The payment gateway checks the contents of the payment response (PaRES).
    a. If the transaction is declined (following a 3D Secure authentication failure), move to step 11.
    b. If not, the transaction moves to step 10.
10. The payment gateway then submits the transaction to the bank for authorisation. The results of the transaction are then passed back to the merchant's system (CallbackURL) in a transparent redirect. The data passed to the Merchant's System should be checked for errors at this point.

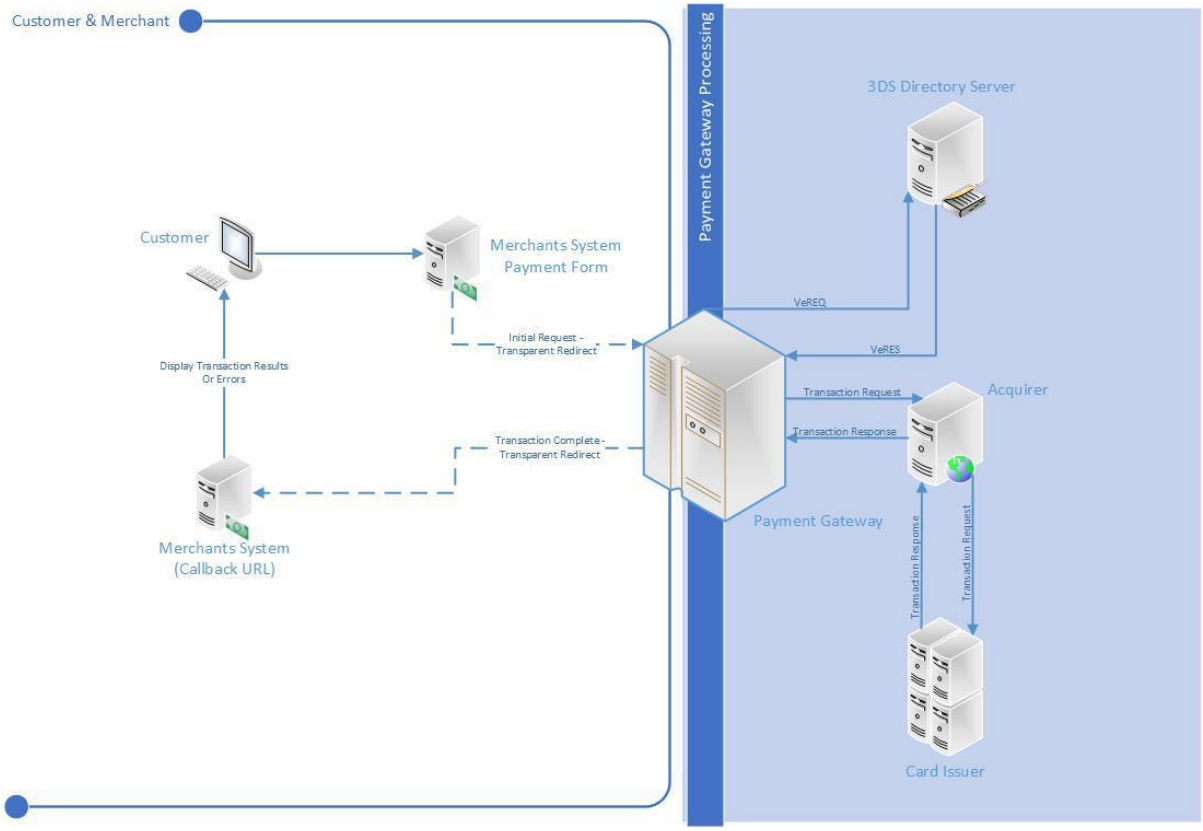
   a. If errors occur (for example; Variable Tampering), the transaction **HAS already been** processed, but the merchant's system should stop the transaction from going any further.

11. The merchant's system should display the transaction result to the customer (or desired error information if any occurred before this point)

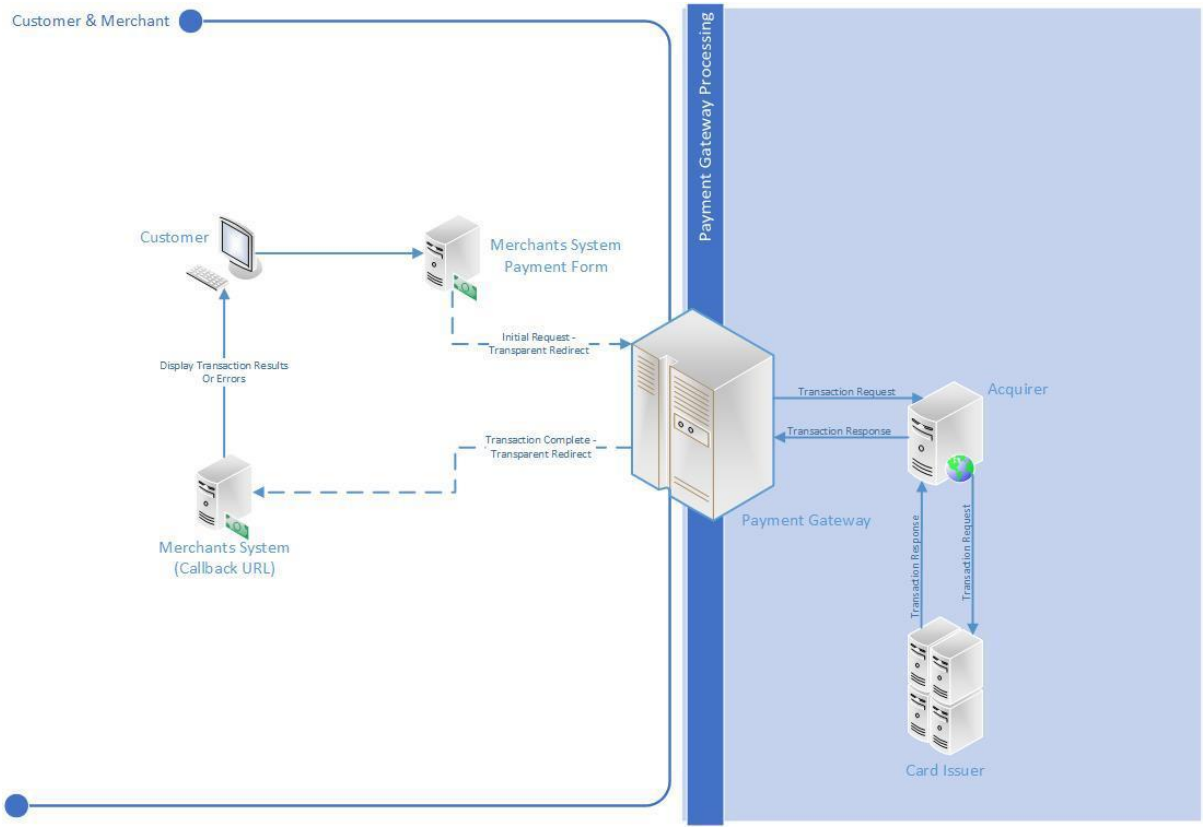
Transaction Flow Maps - Including 3D Secure Authentication



Transaction Flow Maps - 3D Secure Card Not Enrolled



Transaction Flow Maps - 3D Secure Disabled (MMS Setting)



ACS Simulator

The test system comes complete with an ACS simulator, which allows your developer to simulate the most common responses that might come back from the cardholder's bank's access control server

There are 4 possible conditions that can be simulated:

1. Password Correct – the case where the cardholder enters the correct 3D Secure password. Relates to a 3D Secure status of "Y"
2. Password Incorrect – the case where the cardholder enters the wrong 3D Secure password. Relates to a 3D Secure status of "N"
3. Attempted Processing – the case where the cardholder attempted to authenticate themselves, but this could not be completed for some reason. Proof of this attempt is returned with the payment response message. Relates to a 3D Secure status of "A"
4. Unknown Error – the case where an unexpected error occurred whilst trying to authenticate the cardholder. Relates to a 3D Secure status of "U"

Appendix 3: Code examples

StringToHash

This is a sample SALE transaction StringToHash for the transparent redirect URL, this is used to generate the HashDigest.

```
MerchantID=**MERCHANTID**&Password=**PASSWORD**&Amount=9863&CurrencyCode=826&EchoAVSCheckResult=true&EchoCV2CheckResult=true&EchoThreeDSecureAuthenticationCheckResult=true&EchoCardType=true&OrderID=Order-157&TransactionType=SALE&TransactionDateTime=2019-11-07 09:30:16 +00:00&CallbackURL=http://gateway.test/callback.php&OrderDescription=Order description
```

takepayments
we make it easy

Request Form

This is a sample SALE transaction HTML form which is submitted to the gateway transparent redirect page.

```
<form method="POST"
action="https://mms.tponlinepayments.com/Pages/PublicPages/TransparentRedirect.aspx">
    <input type="hidden" name="MerchantID" value="**MERCHANTID**">
    <input type="hidden" name="HashDigest"
value="21859fd704e7e100fc56f59815a2eefbd8d5a765">
    <input type="hidden" name="TransactionType" value="SALE">
    <input type="hidden" name="CallbackURL" value="http://gateway.test/callback.php">
    <input type="hidden" name="Amount" value="9863">
    <input type="hidden" name="CurrencyCode" value="826">
    <input type="hidden" name="OrderID" value="Order-157">
    <input type="hidden" name="TransactionDateTime" value="2019-11-07 09:30:16 +00:00">
    <input type="hidden" name="OrderDescription" value="Order description ">
    <input type="hidden" name="CardName" value="Geoff Wayne">
    <input type="hidden" name="CardNumber" value="4976350000006891">
    <input type="hidden" name="CV2" value="341">
    <input type="hidden" name="ExpiryDateMonth" value="01">
    <input type="hidden" name="ExpiryDateYear" value="25">
    <input type="hidden" name="Address1" value="113 Broad Street West">
    <input type="hidden" name="Address2" value="">
    <input type="hidden" name="Address3" value="">
    <input type="hidden" name="Address4" value="">
    <input type="hidden" name="City" value="Oldpine">
    <input type="hidden" name="State" value="Strongbarrow">
    <input type="hidden" name="PostCode" value="SB42 1SX">
    <input type="hidden" name="CountryCode" value="826">
    <input type="hidden" name="EchoAVSCheckResult" value="true">
    <input type="hidden" name="EchoCV2CheckResult" value="true">
    <input type="hidden" name="EchoThreeDSecureAuthenticationCheckResult" value="true">
    <input type="hidden" name="EchoCardType" value="true">
</form>
```

3D Secure Authentication Required – Response

This is an example response for a transaction requiring 3D Secure Authentication. This response would have been sent via POST to the merchant's CallbackURL from the gateway. This contains a HashDigest that can be used to verify that the response is as expected.

```
HashDigest: d6b88898177d86806a322e06f0f3fa544535a3bc
MerchantID: **MERCHANTID**
StatusCode: 3
Message: Issuer authentication required
CrossReference: 191107093208704502245361
OrderID: Order-157
TransactionDateTime: 2019-11-07 09:30:16 +00:00
ACSURL: https://gw2.tponlinepayments2.com:4430/ACS/Default.aspx
PaREQ:
eJxVUttOg0AQfTfxHwjvdlkoFJphmyoaG9NLlD74uC4biimXLmCKX+8sBavJJjvnzGVnziwszvnR+JKqzs
oiNOnEMg1ZiDLJijQ09/HTnW8u2O0NxAclZfQmRaskg7Wsa55KI0tCc7d8laepY1PPnTqWbzLoGQZDVY
ZFJzaQEWKyEgdeNAy4ON2vNozaztT1Zj6QgYBcqlX0ywdALgQUPJdsx7vvspDG/gVIT4Ao26JRHfNtD8g
IoFVHdmiaak5IdUlp8w/FRQlEu4BcG9m12qqx1DlL2DZK6TpadZv4He/Hbv2ZOtt4TxGHQHQEJLyRzLZ
oQKk1M6xg7thzCwfoeeC57oEF/sRzcKgLgko/shxd2vOXAVRWofTjFCMCea6wdYxADX9tSGQt2FYlUhn
aVFnVoLgGdqAdQK4TPTxroUWD+uGG8FDHcy2tdc/p+hnKZbsW7R/QAIjOIsMiybButP59gx8sVrsr
```

3D Secure Authentication Required - StringToHash

This is a sample StringToHash for transaction requiring 3D Secure Authentication, this is used to generate the HashDigest.

```
MerchantID=**MERCHANTID**&Password=**PASSWORD** &StatusCode=3&Message=Issuer
authentication required&CrossReference=191107093208704502245361&OrderID=Order-
157&TransactionDateTime=2019-11-07 09:30:16 +00:00&ACSURL=https://gw2.
tponlinepayments2.com:4430/ACS/Default.aspx&PaREQ=eJxVUttOg0AQfTfxHwjvdlkoFJphmyoa
G9NLlD74uC4biimXLmCKX+8sBavJJjvnzGVnziwszvnR+JKqzsoiNOnEMg1ZiDLJijQ09/HTnW8u2O0N
xAclZfQmRaskg7Wsa55KI0tCc7d8laepY1PPnTqWbzLoGQZDVYZFJzaQEWKyEgdeNAy4ON2vNozazt
T1Zj6QgYBcqlX0ywdALgQUPJdsx7vvspDG/gVIT4Ao26JRHfNtD8gIoFVHdmiaak5IdUlp8w/FRQlEu4B
cG9m12qqx1DlL2DZK6TpadZv4He/Hbv2ZOtt4TxGHQHQEJLyRzLZoQKk1M6xg7thzCwfoeeC57oEF/
sRzcKgLgko/shxd2vOXAVRWofTjFCMCea6wdYxADX9tSGQt2FYlUhnaVFnVoLgGdqAdQK4TPTxroU
WD+uGG8FDHcy2tdc/p+hnKZbsW7R/QAIjOIsMiybButP59gx8sVrsr
```

3D Secure Authentication Required – Request

Using the information sent from the gateway a Request is sent to the ACSURL to process and display the 3D Secure authentication screen, the TermURL is used at the CallbackURL once the customer has completed 3D secure authentication.

```
<form method="POST" action="https://gw2.tponlinepayments2.com:4430/ACS/Default.aspx"
target="ACSFrame">
    <input type="hidden" name="PaReq"
value="eJxVUttOg0AQfTfxHwjvdlkoFJphmyoaG9NLlD74uC4biimXLmCKX+8sBavJJjvnzGVnziwszvnR
+JKqzsoiNOnEMg1ZiDLJijQ09/HTnW8u2O0NxAclZfQmRaskg7Wsa55KI0tCc7d8laepY1PPnTqWbzLoG
QZDVYZFJzaQEWKyEgdeNAy4ON2vNozaztT1Zj6QgYBcqlX0ywdALgQUPJdsx7vvspDG/gVIT4Ao26JRH
fNtD8gloFVHdmiaak5IdUlp8w/FRQlEu4BcG9m12qqx1DlL2DZK6TpadZv4He/Hbv2ZOtt4TxGHQHQEJ
LyRzLZoQKk1M6xg7thzCwfoeeC57oEF/sRzcKgLgko/shxd2vOXAVRWofTjFCMCea6wdYxADX9tSGQt2
FYlUhnaVFnVoLgGdqAdQK4TPTxroUWD+uGG8FDHcy2tdc/p+hnKZbsW7R/QAIjOIsMiybButP59gx8sVr
sr">
    <input type="hidden" name="MD" value="191107093208704502245361">
    <input type="hidden" name="TermUrl" value="http://gateway.test/callback.php">
</form>
```

3D Secure Authentication Completed – Response

This is an example response for a transaction requiring 3D Secure Authentication. This response would have been sent via POST to the merchant's TermUrl from the gateway.

```
PaRes:
eJxdkttuwjAMhu8n7R2q3o8cekYmiKkDMQmGtqIBd6WNSid6oAcEbz83Y8CWm9hfHPuPHRiesr12lFW
dFvlAZz2qazKPijjNk4G+DMZPrj4Ujw8Q7Cop/Q8ZtZUUMJN1HSZSS+OBvhi9y4NpcGZbpkFdXUBHanW
WspJVDm+4ifhSRWCRHgfy62KyKtqFeSMgjA7P07lg3DAt23GBXABkspr6V+4B+QFAbncXbWfVKO6UxuL
NT9jMn57nwRr3l/PsKzHegiVDfwCki4A4bKTglHmMUUejXt/gfYo1FYeySzfKihZze65tALkngG2osE9n4X
IbyNUDeSqLXGIEPvBqA7mJK8NcmJ5jGxZVy3Y9hrmRQrAS0KTZP1Fm36JAFIe6CZu2FmsgFwui8HgUm
9Vru52U9Xayt9fGuNhM5jzizSH6NPGxKgRklApqoSjc1a3RPimqtNllndS/AEgnhagxqh6rYXf4/hN8A+Rht
ZU=
```

3D Secure Authentication Completed - StringToHash

This is a sample StringToHash for transaction requiring 3D Secure Authentication, this is used to generate the HashDigest.

```
MerchantID=**MERCHANTID**&Password=**PASSWORD**&CrossReference=191107093208704502245
361&TransactionDateTime=2019-11-07 09:34:54
+00:00&CallbackURL=http://gateway.test/callback.php&PaRES=eJxdkttuwjAMhu8n7R2q3o8cekY
miKkDMQmGtqIBd6WNSid6oAcEbz83Y8CWm9hfHPuPHRiesr12lFWdFvlA
```

takepayments
we make it easy

3D Secure Authentication Completed – Request

Using the information sent from the gateway a Request is sent to the Transparent Redirect URL to process, please note that the MD variables becomes the CrossReference

```
<form  method="POST"
action="https://mms.tponlinepayments2.com/Pages/PublicPages/TransparentRedirect.aspx"
target="_parent">
    <input type="hidden" name="PaRES"
value="eJxdkttuwjAMhu8n7R2q3o8cekYmiKkDMQmGtqlBd6WNSid6oAcEbz83Y8CWm9hfHPuPHRies
r12lFWdFvlAZz2qazKPijjNk4G+DMZPrj4Ujw8Q7Cop/Q8ZtZUUMJN1HSZSS+OBvhi9y4NpcGZbpkFdX
UBHanWWspJVDm+4ifhSRWCRHgfy62KyKtqFeSMgjA7P07lg3DAt23GBXABkspr6V+4B+QFAbncXbWfV
KO6UxuLNT9jMn57nwRr3l/PsKzHegiVDfwCki4A4bKTglHmMUUejXt/gfYo1FYeySzfKihZze65tALkngG
2osE9n4XIbyNUDeSqLXGIEPvBqA7mJK8NcmJ5jGxZVy3Y9hrmRQrAS0KTZP1Fm36JAFIe6CZu2FmsgF
wui8HgUm9Vru52U9Xayt9fGuNhM5jzizSH6NPGxKgRklApqoSjc1a3RPimqtNllndS/AEgnhagxqh6rYXf
4/hN8A+RhtZU=">
    <input type="hidden" name="CrossReference" value="191107093208704502245361">
    <input type="hidden" name="CallbackURL" value="http://gateway.test/callback.php">
    <input type="hidden" name="HashDigest"
value="db1b3aa4cf8373866c0ed8eec667d02ca2547536">
    <input type="hidden" name="MerchantID" value="**MERCHANTID**">
    <input type="hidden" name="TransactionDateTime" value="2019-11-07 09:34:54 +00:00">
</form>
```

Transaction Complete – Response

This is an example response for a completed transaction. This response would have been sent via POST to the merchant's CallbackURL from the gateway.

```
HashDigest: 3534d0b78196b65845f7ae38396f60357d5629a9
MerchantID: **MERCHANTID**
StatusCode: 0
Message: AuthCode: 448178
PreviousStatusCode:
PreviousMessage:
CrossReference: 191107093656165702119743
Amount: 9863
CurrencyCode: 826
OrderID: Order-157
TransactionType: SALE
TransactionDateTime: 2019-11-07 09:36:56 +00:00
OrderDescription: Order description
Address1: 113 Broad Street West
Address2:
Address3:
Address4:
City: Oldpine
State: Strongbarrow
PostCode: SB42 1SX
CountryCode: 250
EmailAddress:
PhoneNumber:
AddressNumericCheckResult: PASSED
PostCodeCheckResult: PASSED
CV2CheckResult: PASSED
ThreeDSecureAuthenticationCheckResult: PASSED
FraudProtectionCheckResult:
CardType: VISA
CardClass: PERSONAL
CardIssuer: CREDIT INDUSTRIEL ET COMMERCIA
CardIssuerCountryCode: 250
```

Transaction Complete - StringToHash

This is a sample StringToHash for a complete transaction to verify that the transaction contains the information expected.

```
MerchantID=**MERCHANTID**&Password=**PASSWORD**&StatusCode=0&Message=AuthCode:
448178&PreviousStatusCode=&PreviousMessage=&CrossReference=191107093656165702119743&A
ddressNumericCheckResult=PASSED&PostCodeCheckResult=PASSED&CV2CheckResult=PASSED&Thr
eeDSecureAuthenticationCheckResult=PASSED&CardType=VISA&CardClass=PERSONAL&CardIssuer
=CREDIT INDUSTRIEL ET
COMMERCIA&CardIssuerCountryCode=250&Amount=9863&CurrencyCode=826&OrderID=Order-
157&TransactionType=SALE&TransactionDateTime=2019-11-07 09:36:56
+00:00&OrderDescription=Order description &Address1=113 Broad Street
West&Address2=&Address3=&Address4=&City=Oldpine&State=Strongbarrow&PostCode=SB42
1SX&CountryCode=250&EmailAddress=&PhoneNumber=
```

Appendix 4: Transaction Result Status Codes

The StatusCode is a numerical value representation for the result of the transaction. If any other code is returned not listed here, it should be treated as an error and handled accordingly.

StatusCode	Explanation
0	Transaction Successful
3	Issuer Authentication Required. This means the card is enrolled for 3D Secure Authentication and is required before the transaction can continue.
5	Transaction Declined. You may sometimes get additional information in the Transaction Result Message as to why if we are informed as to why.
20	Duplication Transaction.
30	An Error Occurred.

Appendix 5: Override Policy Codes & Explanations

OverrideAVSPolicy Codes

The OverrideAVSPolicy codes are 4-character codes which instruct the gateway how to handle the AVS checking for that particular transaction.
The first character determines the behaviour when 1 or more of the results of the address numeric or post code check are known.
The second and third characters determine the behaviour when dealing with partial matches – this is where either the address numeric check or the post code check returns partial matches.
The fourth character determines the behaviour when none of the results of the address numeric or the post code check are known.

Character 1 Codes

Character Code	Explanation
E	This code means fail the transaction if either the address numeric check or post code check has failed
B	This code means fail the transaction only if both the address numeric check and the post code checks have failed
A	This code means fail the transaction only if the address numeric check has failed
P	This code means fail the transaction only if the post code check has failed
N	This code means pass the transaction even if both checks have failed

Character 2 Codes

Character Code	Explanation
P	Treat partial address numeric results as passes
F	Treat partial address numeric results as failures

Character 3 Codes

Character Code	Explanation
P	Treat partial post code results as passes
F	Treat partial post code results as failures

Character 4 Codes

Character Code	Explanation
P	This code means pass the transaction if both results of the AVS check are not known
F	This code means fail the transaction if both results of the AVS check are not known

Examples

- **EFFF** – this is the strongest policy & transactions will only pass if both address numeric & post code checks have passed. Partial matches are treated as failures

- **EPFP** – this policy means that transactions will only pass if both the address numeric & post code checks have passed, but if the results of both are unknown, then pass the transaction. Partial address numeric results are treated as passes, but partial post code checks are treated as failures
- **BPPF** – this policy means that the transaction will fail only if both the address numeric and post code checks have failed, but if the results of both are unknown, then fail the transaction. Both address numeric and post code partial results are treated as passes
- **NPPF** – this policy means that the transaction will pass even if the results of the address numeric and post code checks are failed, but if the results are unknown, then fail the transaction (not a recommended policy!) . Both address numeric and post code partial results are treated as passes
- **NPPP** – this is the weakest policy & transactions will pass regardless of the results of the address numeric & post code checks. Both address numeric and post code partial results are treated as passes.

Questions

Q: Why would the results of the AVS check be unknown?

A: The main reasons for the results of the AVS checks being unknown are:

1. The relevant address data was not passed in with the transaction – the address numeric check is carried out across the Address1, Address1, Address3, Address4, City & State fields – if none of them are present, then the state of the address numeric check will be unknown. Similarly, the post code check is carried out of the field PostCode & if that is not present, then the state of the post code check will be unknown.
2. If the transaction is a cross reference transaction & the respective address information was not submitted with the transaction, or was not submitted or unknown for the transaction being referenced, then the result will carry forward to this transaction
3. If there was a problem contacting the provider, or the provider itself had a problem delivering the results of the AVS checks (least likely reason)

OverrideCV2Policy Codes

The OverrideCV2Policy codes are 2-character codes which instruct the gateway how to handle the CV2 checking for that particular transaction.
The first character determines the behaviour when 1 or more of the results of the address numeric or post code check are known.
The second character determines the behaviour when none of the results of the address numeric or the post code check are known.

Character 1 Codes

Code	Explanation
P	This code means pass the transaction if the CV2 check has failed
F	This code means fail the transaction if the CV2 check has failed

Character 2 Codes

Character Code	Explanation
P	This code means pass the transaction if both results of the CV2 check are not known
F	This code means fail the transaction if both results of the CV2 check are not known

Examples

- **FF** – this is the strongest policy & transactions will only pass if the CV2 check has passed
- **FP** – this policy means that transactions will only pass if the CV2 has passed, but if the results are unknown, then pass the transaction
- **PF** – this policy means that the transaction will pass if the CV2 failed, but if the result of the check is unknown, then fail the transaction (not a recommended policy!)
- **PP** – this is the weakest policy & transactions will pass regardless of the results of the CV2 check

Questions

Q: Why would the CV2 result be unknown?

A: The main reasons for the result being unknown are:

1. The CV2 was not submitted with the transaction
2. If the transaction is a cross reference transaction & the CV2 code was not submitted as an override, or was not submitted or unknown for the original transaction being referenced, then that result will carry forward to this transaction
3. If there was a problem contacting the provider, or the provider itself had a problem delivering the results of the CV2 check (least likely reason)

Appendix 6: Country (ISO 3166-1) Codes

ISO Code	Country
826	United Kingdom
840	United States
036	Australia
004	Afghanistan
248	Åland Islands
008	Albania
012	Algeria
016	American Samoa
020	Andorra
024	Angola
660	Anguilla
010	Antarctica
028	Antigua and Barbuda
032	Argentina
051	Armenia
533	Aruba
040	Austria
031	Azerbaijan
044	Bahamas
048	Bahrain
050	Bangladesh
052	Barbados
112	Belarus
056	Belgium
084	Belize
204	Benin
060	Bermuda
064	Bhutan
068	Bolivia
070	Bosnia and Herzegovina
072	Botswana
074	Bouvet Island
076	Brazil
086	British Indian Ocean Territory
096	Brunei Darussalam
100	Bulgaria
854	Burkina Faso
108	Burundi
116	Cambodia
120	Cameroon
124	Canada
132	Cape Verde
136	Cayman Islands
140	Central African Republic
148	Chad
152	Chile
156	China
162	Christmas Island
166	Cocos (Keeling) Islands
170	Colombia
174	Comoros
178	Congo

ISO Code	Country
180	Congo, Democratic Republic of the
184	Cook Islands
188	Costa Rica
384	Côte d'Ivoire
191	Croatia
192	Cuba
196	Cyprus
203	Czech Republic
208	Denmark
262	Djibouti
212	Dominica
214	Dominican Republic
218	Ecuador
818	Egypt
222	El Salvador
226	Equatorial Guinea
232	Eritrea
233	Estonia
231	Ethiopia
238	Falkland Islands (Malvinas)
234	Faroe Islands
242	Fiji
246	Finland
250	France
254	French Guiana
258	French Polynesia
260	French Southern Territories
266	Gabon
270	Gambia
268	Georgia
276	Germany
288	Ghana
292	Gibraltar
300	Greece
304	Greenland
308	Grenada
312	Guadeloupe
316	Guam
320	Guatemala
831	Guernsey
324	Guinea
624	Guinea-Bissau
328	Guyana
332	Haiti
334	Heard Island and McDonald Islands
336	Holy See (Vatican City State)
340	Honduras
344	Hong Kong
348	Hungary
352	Iceland
356	India
360	Indonesia
364	Iran, Islamic Republic of
368	Iraq

ISO Code	Country
372	Ireland
833	Isle of Man
376	Israel
380	Italy
388	Jamaica
392	Japan
832	Jersey
400	Jordan
398	Kazakhstan
404	Kenya
296	Kiribati
408	Korea, Democratic People's Republic of
410	Korea, Republic of
414	Kuwait
417	Kyrgyzstan
418	Lao People's Democratic Republic
428	Latvia
422	Lebanon
426	Lesotho
430	Liberia
434	Libyan Arab Jamahiriya
438	Liechtenstein
440	Lithuania
442	Luxembourg
446	Macao
807	Macedonia, the former Yugoslav Republic of
450	Madagascar
454	Malawi
458	Malaysia
462	Maldives
466	Mali
470	Malta
584	Marshall Islands
474	Martinique
478	Mauritania
480	Mauritius
175	Mayotte
484	Mexico
583	Micronesia, Federated States of
498	Moldova
492	Monaco
496	Mongolia
499	Montenegro
500	Montserrat
504	Morocco
508	Mozambique
104	Myanmar
516	Namibia
520	Nauru
524	Nepal
528	Netherlands
530	Netherlands Antilles
540	New Caledonia
554	New Zealand

ISO Code	Country
558	Nicaragua
562	Niger
566	Nigeria
570	Niue
574	Norfolk Island
580	Northern Mariana Islands
578	Norway
512	Oman
586	Pakistan
585	Palau
275	Palestinian Territory, Occupied
591	Panama
598	Papua New Guinea
600	Paraguay
604	Peru
608	Philippines
612	Pitcairn
616	Poland
620	Portugal
630	Puerto Rico
634	Qatar
638	Reunion Réunion
642	Romania
643	Russian Federation
646	Rwanda
652	Saint Barthélemy
654	Saint Helena
659	Saint Kitts and Nevis
662	Saint Lucia
663	Saint Martin (French part)
666	Saint Pierre and Miquelon
670	Saint Vincent and the Grenadines
882	Samoa
674	San Marino
678	Sao Tome and Principe
682	Saudi Arabia
686	Senegal
688	Serbia
690	Seychelles
694	Sierra Leone
702	Singapore
703	Slovakia
705	Slovenia
90	Solomon Islands
706	Somalia
710	South Africa
239	South Georgia and the South Sandwich Islands
724	Spain
144	Sri Lanka
736	Sudan
740	Suriname
744	Svalbard and Jan Mayen
748	Swaziland
752	Sweden

ISO Code	Country
756	Switzerland
760	Syrian Arab Republic
158	Taiwan, Province of China
762	Tajikistan
834	Tanzania, United Republic of
764	Thailand
626	Timor-Leste
768	Togo
772	Tokelau
776	Tonga
780	Trinidad and Tobago
788	Tunisia
792	Turkey
795	Turkmenistan
796	Turks and Caicos Islands
798	Tuvalu
800	Uganda
804	Ukraine
784	United Arab Emirates
581	United States Minor Outlying Islands
858	Uruguay
860	Uzbekistan
548	Vanuatu
862	Venezuela
704	Viet Nam
92	Virgin Islands, British
850	Virgin Islands, U.S.
876	Wallis and Futuna
732	Western Sahara
887	Yemen
894	Zambia
716	Zimbabwe

Appendix 7: Currency (ISO 4217) Codes

ISO Code	Currency
826	Pound Sterling
840	US Dollar
978	Euro
971	Afghani
12	Algerian Dinar
32	Argentine Peso
51	Armenian Dram
533	Aruban Guilder
36	Australian Dollar
944	Azerbaijanian Manat
44	Bahamian Dollar
48	Bahraini Dinar
764	Baht
590	Balboa
50	Bangladeshi Taka
52	Barbados Dollar
974	Belarusian Ruble
84	Belize Dollar
60	Bermudian Dollar
984	Bolivian Mvdol (Funds code)
68	Boliviano
986	Brazilian Real
96	Brunei Dollar
975	Bulgarian Lev
108	Burundian Franc
124	Canadian Dollar
132	Cape Verde Escudo
136	Cayman Islands Dollar
288	Cedi
952	CFA Franc BCEAO
950	CFA Franc BEAC
953	CFP franc
152	Chilean Peso
963	Code reserved for testing purposes
170	Colombian Peso
174	Comoro Franc
977	Convertible Marks
558	Cordoba Oro
188	Costa Rican Colon
191	Croatian Kuna
192	Cuban Peso
196	Cyprus Pound
203	Czech Koruna
270	Dalasi
208	Danish Krone
807	Denar
262	Djibouti Franc
678	Dobra
214	Dominican Peso
951	East Caribbean Dollar
818	Egyptian Pound
230	Ethiopian Birr

takepayments
we make it easy

ISO Code	Currency
955	European Composite Unit (EURCO)
956	European Monetary Unit
958	European Unit of Account 17 (E.U.A.-17)
957	European Unit of Account 9 (E.U.A.-9)
238	Falkland Islands Pound
242	Fiji Dollar
348	Forint
976	Franc Congolais
292	Gibraltar pound
959	Gold (one Troy ounce)
600	Guarani
324	Guinea Franc
328	Guyana Dollar
332	Haiti Gourde
344	Hong Kong Dollar
980	Hryvnia
352	Iceland Krona
356	Indian Rupee
364	Iranian Rial
368	Iraqi Dinar
388	Jamaican Dollar
392	Japanese yen
400	Jordanian Dinar
404	Kenyan Shilling
598	Kina
418	Kip
233	Kroon
414	Kuwaiti Dinar
894	Kwacha
454	Kwacha
973	Kwanza
104	Kyat
981	Lari
428	Latvian Lats
422	Lebanese Pound
8	Lek
340	Lempira
694	Leone
430	Liberian Dollar
434	Libyan Dinar
748	Lilangeni
440	Lithuanian Litas
426	Loti
969	Malagasy Ariary
458	Malaysian Ringgit
470	Maltese Lira
795	Manat
480	Mauritius Rupee
943	Metical
484	Mexican Peso
979	Mexican Unidad de Inversion (UDI)
498	Moldovan Leu
504	Moroccan Dirham
566	Naira

ISO Code	Currency
232	Nakfa
516	Namibian Dollar
524	Nepalese Rupee
532	Netherlands Antillian Guilder
376	New Israeli Shekel
901	New Taiwan Dollar
949	New Turkish Lira
554	New Zealand Dollar
64	Ngultrum
999	No currency
408	North Korean Won
578	Norwegian Krone
604	Nuevo Sol
478	Ouguiya
776	Pa'anga
586	Pakistan Rupee
964	Palladium (one Troy ounce)
446	Pataca
858	Peso Uruguayo
608	Philippine Peso
962	Platinum (one Troy ounce)
72	Pula
634	Qatari Rial
320	Quetzal
512	Rial Omani
116	Riel
642	Romanian Leu
946	Romanian New Leu
462	Rufiyaa
360	Rupiah
643	Russian Ruble
646	Rwanda Franc
654	Saint Helena Pound
882	Samoan Tala
682	Saudi Riyal
941	Serbian Dinar
690	Seychelles Rupee
961	Silver (one Troy ounce)
702	Singapore Dollar
703	Slovak Koruna
90	Solomon Islands Dollar
417	Som
706	Somali Shilling
972	Somoni
710	South African Rand
410	South Korean Won
960	Special Drawing Rights
144	Sri Lanka Rupee
938	Sudanese Pound
968	Surinam Dollar
752	Swedish Krona
756	Swiss Franc
760	Syrian Pound
834	Tanzanian Shilling

ISO Code	Currency
398	Tenge
780	Trinidad and Tobago Dollar
496	Tugrik
788	Tunisian Dinar
800	Uganda Shilling
970	Unidad de Valor Real
990	Unidades de formento
784	United Arab Emirates dirham
860	Uzbekistan Som
548	Vatu
862	Venezuelan bolívar
704	Vietnamese đồng
947	WIR Euro
948	WIR Franc
886	Yemeni Rial
156	Yuan Renminbi
716	Zimbabwe Dollar
985	Zloty
997	No currency
998	No currency