Virtual Payment Client

Reference Guide

Version 4.2.2

For TNSPay 4.2

Disclaimer

This document is provided by TNS on the basis that you will treat it as confidential. No part of this document may be reproduced or copied in any form by any means without the written permission of TNS. Unless otherwise expressly agreed in writing, the information contained in this document is subject to change without notice and TNS assumes no responsibility for any alteration to, or any error or other deficiency, in this document.

All intellectual property rights in the Document and in all extracts and things derived from any part of the Document are owned by TNS and will be assigned to TNS on their creation. You will protect all the intellectual property rights relating to the Document in a manner that is equal to the protection you provide your own intellectual property. You will notify TNS immediately, and in writing where you become aware of a breach of TNS' intellectual property rights in relation to the Document.

The names “TNS”, any product names of TNS including “Dialect” and “QSI Payments” and all similar words are trademarks of TNS and you must not use that name or any similar name.

TNS may at its sole discretion terminate the rights granted in this document with immediate effect by notifying you in writing and you will thereupon return (or destroy and certify that destruction to TNS) all copies and extracts of the Document in your possession or control.

TNS does not warrant the accuracy or completeness of the Document or its content or its usefulness to you or your merchant cardholders. To the extent permitted by law, all conditions and warranties implied by law (whether as to fitness for any particular purpose or otherwise) are excluded. Where the exclusion is not effective, TNS limits its liability to $100 or the resupply of the Document (at TNS' option). Data used in examples and sample data files are intended to be fictional and any resemblance to real persons or companies is entirely coincidental.

TNS does not indemnify you or any third party in relation to the content or any use of the content as contemplated in these terms and conditions.

Mention of any product not owned by TNS does not constitute an endorsement of that product.

This document is governed by the laws of New South Wales, Australia and is intended to be legally binding.

Contents

Advanced Merchant Administration (AMA) Transactions 83

CHAPTER 1

Preface

Welcome to TNS

TNS Payment Technologies Pty Ltd. ("TNS") is a global provider of payment solutions, connecting merchants and retailers to the world's leading banks, acquirers, and processors, to enable secure, efficient and cost-effective delivery and processing of payments. TNS' payments division provides a wide array of pre-packaged, end-to-end managed solutions designed specifically for the payments industry, enabling customers to focus on their core businesses.

TNS’ Payment Gateway, TNSPay Gateway, is a managed gateway service offering, enabling merchants to authorize and settle card transactions securely, reliably and economically, while ensuring full card data security. TNSPay Gateway is designed to meet the demanding needs of MOTO (mail order/telephone order) merchants and web/eCommerce retailers. Today, TNSPay Gateway represents the platform of choice for over 30,000 merchants, two global card associations, and over 70 banks worldwide. In addition, the solution utilizes our resilient, state-of-the-art global network that transports billions of transactions each year.

For more information on how TNS can help you with your payment processing needs, visit our website at ***http://www.tnsi.com*** http://www.tnsi.com

Audience

This guide is for developers who need to integrate a payments' solution into merchant applications.

Where to Get Help

If you need assistance with the Virtual Payment Client, please contact TNS.

CHAPTER 2

Introduction

TNS' Virtual Payment Client enables merchants to use payment enabled websites, e-commerce or other applications by providing a low effort integration solution. It is suitable for most website hosting environments as merchants can integrate payment capabilities into their application without installing or configuring any payments software.

This guide describes how to payment enable your e-commerce application or on-line store by using the functionality of the Virtual Payment Client.

It details the basic and supplementary fields for the different types of transactions, and includes additional material such as valid codes, error codes and security guidelines.

How This Guide is Structured

This guide consists of the following sections:

Section	Description
Preface	An introduction to TNS and this guide.
Basic Transaction Fields	Details the fields required to perform standard transactions.
Supplementary Transaction Fields	Details the fields required to perform advanced features, for example, Address verification.
AMA Transactions	Details how to setup and perform Advanced Merchant Administration features.
References	Details the valid result field values used by the Payment Server.

Related Documents and Materials

The following material will assist you in your understanding of and implementation of Virtual Payment Client.

Virtual Payment Client Integration Guide

This Virtual Payment Client Reference Guide is designed to be used with the **Virtual Payment Client Integration Guide**. This describes

- how e-Payments work
- describes the various options and models you need to choose before commencing your integration
- describes certain key issues that you must take into account while writing your integration code
- describes the security features available for the Virtual Payment Client, and
- details the various types of transactions of the Virtual Payment Client's API methods.

Merchant Administration User Guide

Merchant Administration allows you to view and manage your electronic transactions through a series of easy to use, secure web pages.

Example code

This is provided by TNS to illustrate the use of the Virtual Payment Client.

Terminology

Term	Description
Access Code	The access code is an identifier that is used to authenticate you as the merchant while you are using the Virtual Payment Client. The access code is generated and allocated to you by Merchant Administrator.
Acquirer Bank	Where your business account is maintained and settlement payments are deposited. This is normally the same bank with which you maintain your merchant facility for your online credit card payments.
Bank	The bank with which you have a merchant facility that allows you to accept online credit card payments.
Capture	A capture is a transaction that uses the information from an authorization transaction to initiate a transfer of funds from the cardholder's account to the merchant's account.
Card Token	The identifier for the stored card details that may be used later to refer to the card details to perform a payment.
Financial Institution (FI)	See Bank.
Issuing Bank	The financial institution that issues credit cards to customers.
Merchant Administration	Merchant Administration allows you to monitor and manage your electronic transactions through a series of easy to use, secure web pages.
Payment Provider	The Payment Provider acts as a gateway between your application or website and the financial institution. It uses the Payment Server to take payment details (Transaction Request) from your cardholder and checks the details with the cardholder's bank. It then sends the Transaction Response back to your application. Approval or rejection of the transaction is completed within seconds, so your application can determine whether or not to proceed with the cardholder's order. Your Payment Provider may be your acquirer bank or a third party technology services provider.
Payment Server	The Payment Server facilitates the processing of secure payments in real-time over the Internet between your application/website and the Payment Provider. All communications between the cardholder, your application, the Payment Server and the Payment Provider is encrypted, making the whole procedure not only simple and quick, but also secure.
Purchase	Purchase is a single transaction that immediately debits the funds from a cardholder's credit card account.
RRN	The RRN (Reference Retrieval Number) is a unique number generated by the payment provider for a specific merchant ID. It is used to retrieve original transaction data and it is useful when your application does not provide a receipt number.
Transaction Request	This is also called the Digital Order (DO) and is a request from the Virtual Payment Client to the Payment Server to provide transaction information.
Transaction Response	This is also called the Digital Receipt (DR) and is a response from the Payment Server to the Virtual Payment Client to indicate the outcome of the transaction.

Virtual Payment Client	The Virtual Payment Client is the interface that provides a secure method of communication between your application and the Payment Server, which facilitates the processing of payments with your financial institution. It allows a merchant application to directly connect using HTTPS protocol in the merchant's choice of programming language.
Transaction	A combination of a Transaction Request and a Transaction Response. For each customer purchase or order, merchants may issue several transactions.

CHAPTER 3

Basic Transaction Fields

This section describes the commands, field types and valid values for basic transactions in Virtual Payment Client.

Field Types

Virtual Payment Client uses 3 different types of fields; *Alpha*, *Alphanumeric* and *Numeric* as described in the table below.

Field Types	Description
Alpha	Alphabetical characters only, in the range **A** to **Z** and **a** to **z** of the base US ASCII characters. The US ASCII ranges for these characters are decimal 65 to 90 inclusive, and decimal 97 to 122 inclusive.
Alphanumeric	Any of the base US ASCII characters in the range decimal 20 to 126 except the | character, decimal 124.
Numeric	Numeric characters only in the range **0** to **9** in the base US ASCII characters. The US ASCII ranges for these characters are decimal 48 to 57 inclusive.

Input Requirements

The Virtual Payment Client requires a number of inputs to perform a basic transaction. The values of these inputs are passed from the merchant software into the Payment Server via the Virtual Payment Client interface.

Depending on the model, 2-Party or 3-Party, the appropriate suffix must be appended to the Virtual Payment Client URL, https://VPC_URL

2-Party Payment Model

The 2-Party Payment Model can be used for any payment application, except where 3-D Secure Authentication is required.

- Data is sent via HTTP POST to https://VPC_URL/vpcdps
- Does not support HTTP GET requests

3-Party Payment Model

The 3-Party Payment Model can be only used for payments where a web browser is involved.

- Data is sent via HTTP GET or POST to https://VPC_URL/vpcpay
- Supports either HTTP GET or POST requests. POST must be used when sensitive data is present in the request. This includes one or more of the following fields:
  - vpc_CardNum
  - vpc_CardSecurityCode
  - vpc_CardTrack1
  - vpc_CardTrack2
  - vpc_User
  - vpc_Password

Note: Sensitive data must never form part of the URI for HTTP GET or POST requests. It must always be sent via POST parameters. A failure to conform to this rule will result in a HTTP Responde code of 400 (Bad Request), and the transaction will fail to proceed.

Input Fields for Basic 2-Party Transactions

Data is sent from the merchant application to the Payment Server via the Virtual Payment Client, a basic transaction requiring a number of data fields as per the table below.

A fully qualified URL (starting with HTTPS://), must be included in the merchant's application code to send transaction information to the Virtual Payment Client.
https://<YOUR_VPC_URL>/vpcdps

Note: This URL is supplied by the Payment Provider.

Base 2-Party Input Fields			
The following data fields must be included in a Transaction Request when using a 2-Party transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Version			
The version of the Virtual Payment Client API being used. The current version is 1.			
Required	Alphanumeric	1,8	1

vpc_Command			
Indicates the desired operation to be performed. This must be equal to '**pay**'.			
Required	Alphanumeric	1,16	pay

vpc_AccessCode			
Authenticates the merchant on the Payment Server. This means that a merchant cannot access another merchant's Merchant Id. The access code is provided when the merchant profile is registered with a Payment Provider.			
Required	Alphanumeric	8	6AQ89F3

vpc_MerchTxnRef

A unique value created by the merchant.

Usage Notes: The Merchant Transaction Reference is used as a reference key to the Payment Server database to obtain a copy of lost/missing transaction receipts using the QueryDR function. It can also be used to identify a duplicate transaction if it is always kept unique for each transaction attempt. It can contain similar information to the vpc_OrderInfo field, but it must be unique for each transaction attempt if it is to be used properly.

Typically, the vpc_MerchTxnRef is based on an order number, invoice number, timestamp, etc., but it should also reflect the transaction attempt. For example, if a cardholder has insufficient funds on their card and they are allowed to repeat the transaction with another credit card, the value may be INV1234/1 on the first attempt, INV1234/2 on the second attempt, and INV1234/3 on the third attempt.

This identifier will be displayed in the Transaction Search results and also in the Download file (from Financial Transactions Search or Download Search Results link in Financial Transaction List) in the Merchant Administration portal on the Payment Server.

Note: If "Enforce Unique Merchant Transaction Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's transactions.

Required	Alphanumeric	1,40	ORDER958743-1

vpc_Merchant

The unique Merchant Id assigned to a merchant by the Payment Provider. The Merchant ID identifies the merchant account against which settlements will be made.

Required	Alphanumeric	1,16	TESTMERCHANT01

vpc_OrderInfo

The merchant's identifier used to identify the order on the Payment Server. For example, a shopping cart number, an order number, or an invoice number.

This identifier will be displayed in the Transaction Search results in the Merchant Administration portal on the Payment Server.

Note: If 'Enforce Unique Order Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's orders.

Required	Alphanumeric	0,34	ORDER958743

vpc_Amount

The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250.

This value cannot be negative or zero. The maximum valid value is 2147483647.

Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.

Required	Numeric	1,12	1250

vpc_CardNum

The number of the card used for the transaction. The format of the Card Number is based on the Electronic Commerce Modeling Language (ECML) and, in particular, must not contain white space or formatting characters.

Required	Numeric	15,19	5123456789012346

<table>
<tr><td colspan="4">vpc_CardExp</td></tr>
<tr><td colspan="4">The expiry date of the card in the format YYMM. The value must be expressed as a 4-digit number (integer) with no white space or formatting characters. For example, an expiry date of May 2013 is represented as 1305.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>4</td><td>1305</td></tr>
<tr><td colspan="4">vpc_Currency</td></tr>
<tr><td colspan="4">The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only.
The merchant must be configured to accept the currency used in this field. To obtain a list of supported currencies and codes, please contact your Payment Provider.</td></tr>
<tr><td colspan="4"><b>Note</b>: This field is required only if more than one currency is configured for the merchant.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>3</td><td>USD</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_SecureHash</td></tr>
<tr><td colspan="4">A secure hash which allows the Virtual Payment Client to authenticate the merchant and check the integrity of the Transaction Request. Secure hash provides better security to merchants than Access Code.
For more details see <b><i>Generating a Secure Hash</i></b> on page 115 and remember to <b><i>always store the Secure Hash secret securely</i></b> on page 118.</td></tr>
<tr><td colspan="4"><b>Note:</b> The secure secret is provided by the Payment Provider.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>64</td><td>9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2</td></tr>
<tr><td colspan="4">vpc_SecureHashType</td></tr>
<tr><td colspan="4">The type of hash algorithm used to generate the secure hash of the Transaction Request and the Transaction Response.
It is strongly recommended that you generate your secure hash using SHA256 HMAC, in which case vpc_SecureHashType=SHA256
For more details see <b><i>Generating a Secure Hash</i></b> on page 115.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>6</td><td>SHA256</td></tr>
<tr><td colspan="4">vpc_ReturnAuthResponseData</td></tr>
<tr><td colspan="4">Specifies whether the authorisation response data must be included in the Transaction Response.
Valid values for this field are:
Y - indicates that the authorisation response data may be included in the Transaction Response, depending on the card type and acquirer used.
N - indicates that the authorisation response data must not be included in the Transaction Response. This is the default value.
For information on authorisation response data, see <b><i>Authorisation Response Code</i></b> on page 131.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>1</td><td>Y</td></tr>
</table>

Input Fields for Basic 3-Party Transactions

Data is sent from the merchant application to the Payment Server via the Virtual Payment Client, a basic transaction requiring a number of data fields as per the table below.

A fully qualified URL (starting with HTTPS://), must be included in the merchant's application code to send transaction information to the Virtual Payment Client.
https://<YOUR_VPC_URL>/vpcdps

Note: This URL is supplied by the Payment Provider.

Base 3-Party Input Fields			
The following data fields must be included in a Transaction Request when using for a 3-Party transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Version			
The version of the Virtual Payment Client API being used. The current version is 1.			
Required	Alphanumeric	1,8	1

vpc_Command			
Indicates the desired operation to be performed. This must be equal to '**pay**'.			
Required	Alphanumeric	1,16	pay

vpc_AccessCode			
Authenticates the merchant on the Payment Server. This means that a merchant cannot access another merchant's Merchant Id. The access code is provided when the merchant profile is registered with a Payment Provider.			
Required	Alphanumeric	8	6AQ89F3

vpc_MerchTxnRef

A unique value created by the merchant. **Usage Notes:** The Merchant Transaction Reference is used as a reference key to the Payment Server database to obtain a copy of lost/missing transaction receipts using the QueryDR function. It can also be used to identify a duplicate transaction if it is always kept unique for each transaction attempt. It can contain similar information to the vpc_OrderInfo field, but it must be unique for each transaction attempt if it is to be used properly. Typically, the vpc_MerchTxnRef is based on an order number, invoice number, timestamp, etc., but it should also reflect the transaction attempt. For example, if a cardholder has insufficient funds on their card and they are allowed to repeat the transaction with another credit card, the value may be INV1234/1 on the first attempt, INV1234/2 on the second attempt, and INV1234/3 on the third attempt. This identifier will be displayed in the Transaction Search results and also in the Download file (from Financial Transactions Search or Download Search Results link in Financial Transaction List) in the Merchant Administration portal on the Payment Server.			
Note: If "Enforce Unique Merchant Transaction Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's transactions.			
Required	Alphanumeric	1,40	ORDER958743-1

vpc_Merchant			
The unique Merchant Id assigned to a merchant by the Payment Provider. The Merchant ID identifies the merchant account against which settlements will be made.			
Required	Alphanumeric	1,16	TESTMERCHANT01
vpc_OrderInfo			
The merchant's identifier used to identify the order on the Payment Server. For example, a shopping cart number, an order number, or an invoice number. This identifier will be displayed in the Transaction Search results in the Merchant Administration portal on the Payment Server.			
Note: If 'Enforce Unique Order Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's orders.			
Required	Alphanumeric	0,34	ORDER958743

vpc_Amount			
The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.			
Required	Numeric	1,12	1250
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. The merchant must be configured to accept the currency used in this field. To obtain a list of supported currencies and codes, please contact your Payment Provider.			
Note: This field is required only if more than one currency is configured for the merchant.			

Optional	Alpha	3	USD

<table>
<tr><th colspan="4">vpc_Locale</th></tr>
<tr><td colspan="4">Specifies the language used on the Payment Server pages that are displayed to the cardholder, in 3-Party transactions. Please check with your Payment Provider for the correct value to use.
In a 2-Party transaction the default value of 'en' is used.</td></tr>
<tr><td>Required</td><td>Alphanumeric</td><td>2,5</td><td>en</td></tr>
</table>

<table>
<tr><th colspan="4">vpc_ReturnURL</th></tr>
<tr><td colspan="4">URL supplied by the merchant in a 3-Party transaction. It is used by the Payment Server to redirect the cardholder's browser back to the merchant's web site. The Payment Server sends the encrypted Digital Receipt with this URL for decryption.
It must be a fully qualified URL starting with HTTP:// or HTTPS:// and if typed into a browser with Internet access, would take the browser to that web page.
It is recommended that the browser is returned to an SSL secured page. This will prevent the browser pop-up indicating that the cardholder is being returned to an unsecure site. If the cardholder clicks 'No' to continue, then neither the merchant or the cardholder will obtain any receipt details.</td></tr>
<tr><td>Required</td><td>Alphanumeric</td><td>1,255</td><td>https://merchants_site/receipt.asp</td></tr>
<tr><th colspan="4">vpc_SecureHash</th></tr>
<tr><td colspan="4">A secure hash which allows the Virtual Payment Client to authenticate the merchant and check the integrity of the Transaction Request. Secure hash provides better security to merchants than Access Code.
For more details see <b><i>Generating a Secure Hash</i></b> on page 115 and remember to <b><i>always store the Secure Hash secret securely</i></b> on page 118.
<b>Note:</b> The secure secret is provided by the Payment Provider.</td></tr>
<tr><td>Required</td><td>Alphanumeric</td><td>64</td><td>9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2</td></tr>
<tr><th colspan="4">vpc_SecureHashType</th></tr>
<tr><td colspan="4">The type of hash algorithm used to generate the secure hash of the Transaction Request and the Transaction Response.
It is strongly recommended that you generate your secure hash using SHA256 HMAC, in which case vpc_SecureHashType=SHA256
For more details see <b><i>Generating a Secure Hash</i></b> on page 115.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>6</td><td>SHA256</td></tr>
<tr><th colspan="4">vpc_ReturnAuthResponseData</th></tr>
<tr><td colspan="4">Specifies whether the authorisation response data must be included in the Transaction Response.
Valid values for this field are:
Y - indicates that the authorisation response data may be included in the Transaction Response, depending on the card type and acquirer used.
N - indicates that the authorisation response data must not be included in the Transaction Response. This is the default value.
For information on authorisation response data, see <b><i>Authorisation Response Code</i></b> on page 131.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>1</td><td>Y</td></tr>
</table>

Basic Output Fields

Once a Transaction Response has been successfully received, the merchant application can retrieve the receipt details. These values are then passed back to the cardholder for their records.

Note: The Transaction Response provided by the Payment Server may contain other fields that are not documented in this guide. Such fields may be changed, added, or removed without notice, and must NOT be relied upon by merchant integrations.

Terminology: Returned Input fields are shown as "Input" in the table.

Base Output Fields			
The following data fields are returned in a Transaction Response for standard 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
The value of the vpc_Command input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	pay

vpc_MerchTxnRef			
The value of the vpc_MerchTxnRef input field returned in the Transaction Response. This field may not be returned in a transaction that fails due to an error condition.			
Input	Alphanumeric	0,40	ORDER958743-1

vpc_Merchant			
The value of the vpc_Merchant input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	TESTMERCHANT01

vpc_OrderInfo			
The value of the vpc_OrderInfo input field returned in the Transaction Response.			
Input	Alphanumeric	1,34	ORDER958743

vpc_Amount			
The value of the vpc_Amount input field returned in the Transaction Response.			
Input	Numeric	1,10	1250
vpc_Currency			
The value of the vpc_Currency input field returned in the Transaction Response. This field is returned only if vpc_Currency was included in the Transaction Request.			
Input	Alpha	3	USD
vpc_Message			

This is a message to indicate what sort of errors the transaction encountered. This field is not provided if vpc_TxnResponseCode has a value of zero.			
Output	Alphanumeric	1,255	Merchant [TESTCORE23] does not exist.

vpc_TxnResponseCode			
A response code that is generated by the Payment Server to indicate the status of the transaction. A vpc_TxnResponseCode of “0” (zero) indicates that the transaction was processed successfully and approved by the acquiring bank. Any other value indicates that the transaction was declined (it went through to the banking network) or the transaction failed (it never made it to the banking network). For a list of values, see Transaction Response Codes.			
Output	Alphanumeric	1	0
vpc_ReceiptNo			
A unique identifier that is also known as the Reference Retrieval Number (RRN). The vpc_ReceiptNo may be passed back to the cardholder for their records if the merchant application does not generate its own receipt number. This field is not returned for transactions that result in an error condition.			
Output	Alphanumeric	0,12	RP12345
vpc_AcqResponseCode			
Generated by the financial institution to indicate the status of the transaction. The results can vary between institutions so it is advisable to use the vpc_TxnResponseCode as it is consistent across all acquirers. It is only included for fault finding purposes. Most Payment Providers return the vpc_AcqResponseCode as a 2-digit response, others return it as a 3-digit response. This field is not returned for transactions that result in an error condition.			
Output	Alphanumeric	2,3	00

vpc_TransactionNo			
Payment Server OrderID (or Shopping Transaction Number) is a unique number generated by the Payment Server for every transaction. It is important to ensure that the vpc_TransactionNo is stored for later retrieval. It is used in Merchant Administration and Advanced Merchant Administration as a reference to perform refund, capture and void transactions. This field is not returned for transactions that result in an error condition.			
Output	Numeric	1,19	96841

vpc_BatchNo			
A value supplied by an acquirer which indicates the batch of transactions that the specific transaction has been grouped with. Batches of transactions are settled by the acquirer at intervals determined by them. This is an acquirer specific field, for example, it could be a date in the format YYYYMMDD. This field will not be returned if the transaction fails due to an error condition.			
Output	Numeric	0,8	20060105

vpc_AuthorizeId			
Authorisation Identification Code issued by the Acquirer to indicate the approval of a transaction. This field is 6-digits maximum and is not returned for transactions that are declined or fail due to an error condition. **Note**: This field may not be returned based on the transaction type and your acquirer configuration.			
Output	Alphanumeric	0,6	654321

vpc_Card			
Identifies the card type used for the transaction. For a list of card types see Card Type Codes. This field is not returned for transactions that result in an error condition.			
Output	Alpha	0,2	MC

vpc_SecureHash			
Allows the merchant application to check the integrity of the returning Transaction Response. ***Always store the Secure Hash secret securely*** on page 118.			
Output	Alphanumeric	64	9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2
vpc_SecureHashType			
The value of vpc_SecureHashType returned in the Transaction Response.			
Input	Alphanumeric	6	SHA256
vpc_ReturnACI			
The ACI (Authorisation Characteristics Indicator) returned by the issuer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	1	1
vpc_TransactionIdentifier			
The unique identifier for the transaction returned by the issuer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	0, 19	ABC187659DEFGJ0
vpc_CommercialCardIndicator			
Indicates the type of commercial card as returned by the card issuer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	1	B
vpc_CommercialCard			
Indicates if the card used is a commercial card. For more information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	1	Y
vpc_CardLevelIndicator			

Indicates the card level result returned by the issuer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	2	A [Character "A" followed by a space]
vpc_FinancialNetworkCode			
Indicates the code of the financial network that was used to process the transaction with the issuer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	0,3	AB2
vpc_MarketSpecificData			
Indicates the market or the industry associated with the payment. For example, B and H may indicate "bill payment" and "hotel" respectively depending on the acquirer. For information, see ***Authorisation Response Code*** on page 131.			
Note: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	0,1	A

CHAPTER 4

Supplementary Transaction Fields

The following sections detail the additional functionality available to merchants. The base fields for either 2-Party or 3-Party transactions are used with the extra fields detailed in these sections.

Most functionality is available to both 2-Party and 3-Party transactions, some are limited to only 2-Party or 3-Party, but are designated as such in the details.

Note: While these are supplementary fields, some of these fields may be mandatory for certain functions.

Address Verification Service (AVS) Fields

The Address Verification Service (AVS) is a security feature used for card not present transactions. It compares the card billing address data that the cardholder supplies with the records held in the card issuer's database. Once the transaction is successfully processed and authorised, the card issuer returns a result code (AVS result code) in its authorisation response message. The result code verifies the AVS level of accuracy used to match the AVS data.

In a standard 3-Party transaction, the merchant does not have to send the AVS data as the Payment Server prompts the cardholder for the information. However, in a 2-Party transaction or 3-Party with card details transaction, the AVS data must be sent by the merchant, if AVS is required.

Note: Applies to 2-Party transactions and 3-Party with card details transactions.

Transaction Request Input Fields

<table>
<tr><th colspan="4">Address Verification Service (AVS) Input Fields</th></tr>
<tr><td colspan="4">The data is sent by simply including the additional data with the required fields for a basic transaction.</td></tr>
<tr><td colspan="4">Field Name</td></tr>
<tr><td colspan="4">Field Description</td></tr>
<tr><td>Required/ Optional</td><td>Field Type</td><td>Min, Max or Set Field Length</td><td>Sample Data</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_AVS_Street01</td></tr>
<tr><td colspan="4">The street name and number, or the Post Office Box details, of the address used in the credit card billing Address Verification check by the card issuing bank.</td></tr>
<tr><td>Required</td><td>Alphanumeric</td><td>1,128</td><td>1136 John Street</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_AVS_City</td></tr>
<tr><td colspan="4">The city/town/village of the address used in the credit card billing Address Verification check by the card issuing bank.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>1,128</td><td>Seattle</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_AVS_StateProv</td></tr>
<tr><td colspan="4">The State/Province code of the address used in the credit card billing Address Verification check by the card issuing bank.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>0,128</td><td>WA</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_AVS_PostCode</td></tr>
<tr><td colspan="4">The Postal/Zip code of the address used in the credit card billing Address Verification check by the card issuing bank.</td></tr>
<tr><td>Required</td><td>Alphanumeric</td><td>4,9</td><td>98111</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_AVS_Country</td></tr>
<tr><td colspan="4">The 3 digit ISO standard alpha country code of the address used in the credit card billing Address Verification check by the card issuing bank.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>3</td><td>USA</td></tr>
</table>

Transaction Response Output Fields

<table>
<tr><th colspan="4">Address Verification Service (AVS) Output Fields</th></tr>
<tr><td colspan="4">In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.</td></tr>
<tr><td colspan="4">Field Name</td></tr>
<tr><td colspan="4">Field Description</td></tr>
<tr><td>Returned Input or Output</td><td>Field Type</td><td>Min, Max or Set Field Length</td><td>Sample Data</td></tr>
</table>

vpc_AVS_Street01			
The value of the vpc_AVS_Street01 input field returned in the Transaction Response.			
Input	Alphanumeric	0,20	1136 John Street

vpc_AVS_City			
The value of the vpc_AVS_City input field returned in the Transaction Response.			
Input	Alphanumeric	0,20	Seattle

vpc_AVS_StateProv			
The value of the vpc_AVS_StateProv input field returned in the Transaction Response.			
Input	Alphanumeric	0,5	WA

vpc_AVS_PostCode			
The value of the vpc_AVS_PostCode input field returned in the Transaction Response.			
Input	Alphanumeric	0,9	98111

vpc_AVS_Country			
The value of the vpc_AVS_Country input field returned in the Transaction Response.			
Input	Alpha	0,3	USA

vpc_AVSResultCode			
The result code generated by the Payment Sever to indicate the AVS level that was used to match the data held by the cardholder's issuing bank. For more information, see ***AVS Result Codes*** on page 121. **Note:** It can also be returned as '**Unsupported**' if the acquirer does not support this field.			
Output	Alpha	1,11	Y

vpc_AcqAVSRespCode			
Generated by the card issuing institution in relation to AVS. Provided for ancillary information only.			
Output	Alpha	1,11	Y

Advanced Address Verification (AAV) Data

The Address Verification Service (AVS) is a security feature used for card not present transactions. It compares the card billing AVS data that the cardholder supplies with the records held in the card issuer's database. Once the transaction is successfully processed and authorised, the card issuer returns a result code (AVS result code) in its authorisation response message. The result code verifies the AVS level of accuracy used to match the AVS data.

In a standard 3-Party transaction, the merchant does not have to send the basic AVS data of vpc_AVS_Street01, vpc_AVS_City, vpc_AVS_StateProv, vpc_AVS_PostCode, vpc_AVS_Country, vpc_BillTo_Firstname, and vpc_BillTo_Lastname as the Payment Server prompts the cardholder for the information. However, in a 2-Party transaction, all the AVS data and its extended fields will need to be sent, if AVS is required.

Note: The merchant can enforce the Card Holder name entry by enabling the *Enforce Card Holder Name entry for 3-Party* privilege in Merchant Manager.

Merchants are encouraged to

- include shipping data on all shipments AND
- to use the 205-Byte format to include shipping data on all shipments,

even if Card member Billing and Ship-to addresses are identical, because this data enhances the ability to assess risk.

Note: Applies to both 2-Party and 3-Party transactions.

Transaction Request Input Fields

Address Verification Service (AVS) Input Fields			
Data is sent from the merchant application to the Payment Server via the Virtual Payment Client, a basic transaction requiring a number of data fields as per the table below.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_AVS_Street01			
The street name and number, or the Post Office Box details, of the address used in the credit card billing Address Verification check by the card issuing bank.			
Required	Alphanumeric	1,128	1136 John Street

vpc_AVS_City			
The city/town/village of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alphanumeric	1,128	Seattle

vpc_AVS_StateProv			

The State/Province code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alphanumeric	0,128	WA

vpc_AVS_PostCode			
The Postal/Zip code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Required	Alphanumeric	4,9	98111

vpc_AVS_Country			
The 3 digit ISO standard alpha country code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alpha	3	USA

vpc_BillTo_Title			
The title of the person that the bill is being sent to.			
Optional	Alphanumeric	2,3	Mr

vpc_BillTo_Firstname			
The first name of the person that the bill is being sent to.			
Note: A 3-party transaction allows a maximum of 50 characters for this field.			
Optional	Alphanumeric	1,15	Alan

vpc_BillTo_Middlename			
The first initial of the middle name of the person that the bill is being sent to. If there is more than one middle name, this is the first middle name.			
Optional	Alphanumeric	0,1	H

vpc_BillTo_Lastname			
The last name or surname of the person that the bill is being sent to.			
Note: A 3-party transaction allows a maximum of 50 characters for this field.			
Optional	Alphanumeric	1,30	Jones

vpc_BillTo_Phone			
The phone number of the person that the bill is being sent to.			
Optional	Numeric	1,10	9876543210

vpc_ShipTo_Title			
The title of the contact person that the current order is being shipped to.			
Optional	Alphanumeric	0,8	Mrs

vpc_ShipTo_Firstname			
The first name of the person that the current order is being shipped to.			
Optional	Alphanumeric	1,15	Jane

vpc_ShipTo_Middlename			
The first initial of the middle name of the person that the current order is being shipped to. If there is more than one middle name, this is the first middle name.			
Optional	Alphanumeric	0,1	Y

vpc_ShipTo_Lastname			
The last name or surname of the person that the current order is being shipped to.			
Optional	Alphanumeric	1,30	Doe

vpc_ShipTo_Phone			
The phone number of the contact person that the current order is being shipped to.			
Optional	Numeric	0,10	5122346788

vpc_ShipTo_Street01			
The street name and number, or the Post Office Box details, of the address that the current order is being shipped to.			
Optional	Alphanumeric	0,60	PO Box 1701

vpc_ShipTo_City			
The 'Ship To' city of the Current Order.			
Optional	Alphanumeric	1,20	Denver

vpc_ShipTo_StateProv			
The state or province to which the current order is being shipped.			
Optional	Alphanumeric	0,20	CO

vpc_ShipTo_PostCode			
The post code or zip code of the address to where the current order is being shipped.			
Optional	Alphanumeric	0,9	213456

vpc_ShipTo_Country			
The 3 digit ISO standard alpha country code of the 'Ship To' address used for the current order.			
Optional	Alpha	3	USA

Transaction Response Output Fields

Address Verification Service (AVS) Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_AVS_Street01			

The value of the vpc_AVS_Street01 input field returned in the Transaction Response.			
Input	Alphanumeric	0,20	1136 John Street

vpc_AVS_City			
The value of the vpc_AVS_City input field returned in the Transaction Response.			
Input	Alphanumeric	0,20	Seattle

vpc_AVS_StateProv			
The value of the vpc_AVS_StateProv input field returned in the Transaction Response.			
Input	Alphanumeric	0,5	WA

vpc_AVS_PostCode			
The value of the vpc_AVS_PostCode input field returned in the Transaction Response.			
Input	Alphanumeric	0,9	98111

vpc_AVS_Country			
The value of the vpc_AVS_Country input field returned in the Transaction Response.			
Input	Alpha	0,3	USA

vpc_AVSResultCode			
The result code generated by the Payment Sever to indicate the AVS level that was used to match the data held by the cardholder's issuing bank. For more information, see ***AVS Result Codes*** on page 121. **Note:** It can also be returned as '**Unsupported**' if the acquirer does not support this field.			
Output	Alpha	1,11	Y

vpc_AcqAVSRespCode			
Generated by the card issuing institution in relation to AVS. Provided for ancillary information only.			
Output	Alpha	1,11	Y

Airline Passenger Data (APD) Fields

The Airline Passenger Data (APD) is a security feature plus added functionality used for card not present transactions. It adds Airline Passenger Data about this transaction.

Note: APD data and Internet Transaction Data (ITD) cannot both be used in a single transaction.

ITD or APD should only be implemented in 2-Party transactions as the payload is too large for a 3-Party style of transaction and the cardholder's browser will not perform a browser redirect.

APD subfields may contain additional travel-specific information, including the departure date, passenger and cardholder names, travel origin and destination, routing cities, airline carriers, fare basis, number of passengers, e-ticket indicator and reservation code.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Airline Passenger Data (APD) Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_APD_DeptDate			
Airline Passenger Data field, Departure Date - Format YYYYMMDD.			
Optional	Numeric	8	20060227

vpc_APD_PassengerTitle			
Passenger's title.			
Optional	Alphanumeric	2,3	Mrs

vpc_APD_PassengerFirstname			
Passenger's first name.			
Optional	Alphanumeric	1,20	Anne

vpc_APD_PassengerMiddlename			
Passenger's middle name (first middle name if more than one).			
Optional	Alphanumeric	1,20	Louise

vpc_APD_PassengerLastname			
Passenger's last name or surname.			
Optional	Alphanumeric	1,20	Smith

vpc_APD_CardmemberTitle			
Cardholder's title.			
Optional	Alphanumeric	2,3	Mr

vpc_APD_CardmemberFirstname			
Cardholder's first name.			
Optional	Alphanumeric	1,20	John

vpc_APD_CardmemberMiddlename			
Cardholder's middle name (first middle name if more than one).			
Optional	Alphanumeric	1,20	Raymond

vpc_APD_CardmemberLastname			
Cardholder's last name or surname.			
Optional	Alphanumeric	1,20	Smith

vpc_APD_Origin			
Airport where the travel starts from. **Note**: Five-byte code sequence allows for anticipated expansion of present, three-character Airport Code.			
Optional	Alpha	3,5	STL

vpc_APD_Dest			
This is the travel destination of the first segment, not necessarily the final destination. For example, if passenger flies from STL to MIA with layover at JFK, Destination Airport for first segment is JFK. **Note**: Five-byte code sequence allows for anticipated expansion of present, three-character Airport Code.			
Optional	Alphanumeric	3,5	JFK

vpc_APD_Route			
Routing Airport or City Codes for each leg of the journey on the ticket (including ORIGIN and DEST) in three-byte segments with virgule (/) separator.			
Optional	Alphanumeric	7,39	ABC/DEF/GHI/JKL/MNO/PQR/STU/VWX/YZA/XYZ

vpc_APD_Carriers			
Airline Carriers - Airline Carrier Code for each leg on ticket (including ORIGIN and DEST) in three-byte segments with virgule (/) separator. **Note**: Each leg must have Airline Carrier Code entry, even if multiple (or all) legs are on the same Airline.			
Optional	Alphanumeric	2,26	AB/XY/BC/CD/DE/DE/CD/BC/AB

vpc_APD_FareBasis			
Primary & secondary discount codes. These indicate the class of service and fare level associated with ticket.			
Optional	Alphanumeric	1,24	ABC123DEF456GHI789JKL012

vpc_APD_NumPassengers			
Number of passengers in party.			

Optional	Numeric	1,3	2

vpc_APD_eTicket			
This flag indicates if ticket is electronic. If a e-Ticket the value is '**Y**' otherwise the value is '**N**'			
Optional	Alpha	1	Y

vpc_APD_ResCode			
Reservation Code (a precursor to a ticket number) corresponds to an airline ticket purchase reservation made by an airline or Global Distribution System (GDS).			
Optional	Alphanumeric	6,15	ABCDE1234567890

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Card Holder Name Fields

This is a security feature in which the Payment Server requests the card holder to provide the card holder name in a standard 3-Party transaction. It may be used to perform fraud checks by comparing the supplied card holder name with the records held in the card issuer's database.

Note: Applies only to 3-Party transactions.

The merchant can enforce the Card Holder name entry by selecting the *Enforce Card Holder Name entry for 3-Party* privilege in Merchant Manager.

Transaction Request Input Fields

Card Holder Name Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_BillTo_Fullname			
The name of the person that the bill is being sent to.			
Optional	Alphanumeric	1, 128	Alan Adam

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Card Security Code (CSC) Field

The Card Security Code (CSC) is a security feature for Card-Not-Present transactions. It is also known as also known as CVV(Visa), CVC2(MasterCard) or CID/4DBC(Amex) or CVV2.

It compares the Card Security Code on the card with the records held in the card issuer's database. For example, on Visa and MasterCard credit cards, it is the three digit value printed on the signature panel on the back following the credit card account number. For American Express, the number is the 4 digit value printed on the front above the credit card account number.

Once the transaction is successfully processed and authorised, the card issuer returns a result code (CSC result code) in its authorisation response message. This verifies the CSC level of accuracy used to match the card security code.

In a standard 3-Party transaction, the merchant does not have to send the Card Security Code as the Payment Server prompts the cardholder for the information. However, in a 2-Party transaction or 3-Party with card details transaction, the merchants application must send the *vpc_CardSecurityCode* value, if CSC is required.

Note: Applies to 2-Party transactions and 3-Party with card details transactions.

Transaction Request Input Fields

Card Security Code (CSC) Input Field			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_CardSecurityCode			
The Card Security Code (CSC), also known as CVV(Visa), CVC2(MasterCard) or CID/4DBC(Amex) or CVV2, which is printed, not embossed on the card. It compares the code with the records held in the card issuing institution's database.			
Optional	Numeric	3,4	985

Transaction Response Output Fields

Output Fields
In addition to the standard output fields, the following field is also returned in the Transaction Response for both 2-Party and 3-Party transactions.
Field Name
Field Description

Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

<table>
<tr><th colspan="4">vpc_CSCResultCode</th></tr>
<tr><td colspan="4">A single digit response from the Payment Server that is mapped from the AcqCSCRespCode showing the level of match that occurred with the CSC check. For more information, see CSC Level Codes.
If the transaction was declined because the CSC check failed, a vpc_TxnResponseCode value of "2" - 'Bank Declined Transaction' will be returned.
If the acquiring institution does not support CSC, the vpc_CSCResultCode will show '<b>Unsupported</b>'.</td></tr>
<tr><td>Output</td><td>Alpha</td><td>1,11</td><td>M</td></tr>
</table>

<table>
<tr><th colspan="4">vpc_AcqCSCRespCode</th></tr>
<tr><td colspan="4">The result code generated by the card issuing institution in relation to the Card Security Code. This is only provided for ancillary information.</td></tr>
<tr><td>Output</td><td>Alpha</td><td>1,11</td><td>M</td></tr>
</table>

External Payment Selection (EPS) Fields

External Payment Selection (EPS) is only used in a 3-Party transaction in order to bypass the Payment Server page that displays the logos of all the available cards that the payment processor accepts. This can be helpful if the merchant's application already allows the cardholder to select the card they want to pay with. This stops the cardholder having to do a double selection, once at the merchant's application and once on the Payment Server.

The first page displayed in the 3-Party Payment process is the card details page for the card type selected.

EPS data is also required to be passed in if the merchant wants to include card details in a 3-Party transaction. The Payment Provider must have set the correct privilege in the Payment Server for EPS to operate.

Note: Applies to 3-Party transactions.

Transaction Request Input Fields

External Payment Selection (EPS) Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Card			
Used in External Payment Selection to determine what type of card is used. The field is case sensitive, and must comply with each of the card types valid in the Payment Server. This varies from Payment Server to Payment Server. The possible values are shown in ***External Payment Selection (EPS)*** on page 123. To check the card types available for your Payment Provider, perform a 3-Party transaction and go to the Payment Server card selection page in a browser. Run the cursor over each card logo.The 'card' and 'gateway' values are displayed at the bottom of the browser window.			
Required	Alphanumeric	3,16	Visa

vpc_Gateway			
Determines the type of payment gateway functionality. The field is case sensitive, and must comply with the gateways that are valid in the Payment Server. Valid values for this field are: ▪ **ssl** — specifies the gateway for all standard 3-Party transactions ▪ **threeDSecure** — specifies the gateway for a 3-D Secure Mode 3a-3-party Style Authentication Only transaction. Note: For most transactions the value of this field will be '**ssl**'			
Required	Alphanumeric	3,15	ssl

vpc_PaymentMethod

Determines the type of payment method or processing network used to process a transaction. The field is case sensitive, and must comply with the payment methods that are valid in the Payment Server.
Valid values for this field are:

- **CREDIT—** specifies the payment method for all standard credit transactions.
- **PAYPAL—** specifies the payment method for a PayPal transaction.

Note: If a valid value is not specified, the payment method defaults to **"CREDIT".**

Optional	Alpha	3,6	CREDIT

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Acquirer Dependent Fields

Acquirer Dependent fields are fields that are dependent on the type of acquirer used to process the transaction. The payment method is set using the vpc_PaymentMethod field. For more information on vpc_PaymentMethod, see ***External Payment Selection (EPS) Fields.*** on page 37

Note: Depending on your acquirer configuration, some fields may be enabled or disabled.

Transaction Request Input Fields

Acquirer Dependent Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_ShipToAddressFromProvider			
Indicates if the Payment Server should request and receive shipping address from the acquirer and return it in the Transaction Response. Valid values are: ▪ Y ▪ N By default, the value for this field is set to N.			
Optional	Alphanumeric	1,1	N

Transaction Response Output Fields

Acquirer Dependent Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for 3-Party transactions, depending on the acquirer used to process the transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_CustomerIdFromProvider			
The identifier received from the acquirer, which is unmasked. For example, for the PayPal Acquirer, the identifier will be the PayPal Payer Email Address.			
Required	Alphanumeric	0,127	scottadam@paypal.com
vpc_AcqResponseText			
The response from the acquirer in the text form. This field is used instead of vpc_AcqResponseCode for acquirers that return text instead of a single code.			

Optional	Alphanumeric	0,255	Success : Pending: Authorization
vpc_ShipTo_FullName			
The full name of the person the current order is being shipped to.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alphanumeric	0,32	Scott Adam
vpc_ShipTo_Street01			
The street name and number, or the Post Office Box details, of the address that the current order is being shipped to.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alphanumeric	0,100	PO Box 1701
vpc_ShipTo_City			
The city that the current order is being shipped to.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alphanumeric	1,40	Denver
vpc_ShipTo_StateProv			
The state or province to which the current order is being shipped.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alphanumeric	0,40	CO
vpc_ShipTo_PostCode			
The post code or zip code of the address to where the current order is being shipped.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alphanumeric	0,20	213456
vpc_ShipTo_Country			
The 3 digit ISO standard alpha country code of the 'Ship To' address used for the current order.			
Note: This field is returned if vpc_ShipToAddressFromProvider is set to "Y".			
Optional	Alpha	3,3	USA
vpc_PaymentMethod			
The value of the vpc_PaymentMethod input field returned in the Transaction Response. vpc_PaymentMethod is the payment method used to process the transaction. This field is returned only if it is included in the Transaction Request.			
Optional	Alpha	3,6	CREDIT

Merchant Transaction Source

This section describes how to use the additional functionality of the Transaction Source field, which allows a merchant to indicate the source of a 2-Party transaction. Merchants and acquirers can optionally set the merchant transaction source so the payment provider can calculate correct fees and charges for each transaction.

Merchant transaction source is added to 2-Party transactions using the supplementary command at the appropriate point as indicated in their transaction flows.

If not specified, this transaction will be set to the merchant's default transaction source.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Merchant Transaction Source Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_TxSource			
Allows the merchant to specify the source of the transaction. Valid Values are: **INTERNET** - indicates an Internet transaction **MOTOCC** - indicates a call centre transaction **MOTO** - indicates a mail order or telephone order **MAILORDER -** indicates a mail order transaction **TELORDER -** indicates a telephone order transaction **CARDPRESENT** - indicates that the merchant has sighted the card. **VOICERESPONSE** - indicates that the merchant has captured the transaction from an IVR system. **Note:** This can only be used if the merchant has *Allow the Merchant to Change the Transaction Source* privilege, otherwise the transaction will be set to the merchant's default transaction source as defined by Transaction Network Services'.			
Optional	Alphanumeric	6,16	INTERNET

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Merchant Transaction Source Frequency

This section describes how use the additional functionality of Transaction Frequency data, which allows a merchant to indicate the frequency of the transaction.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Transaction Source Subtype Field			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_TxSourceSubType			
Allows the merchant to flag the subtype of transaction for the cardholder's order. vpc_TxSourceSubType must be one of the following values: **SINGLE** - indicates a single transaction where a single payment is used to complete the cardholder's order. **INSTALLMENT** - indicates an installment transaction where the cardholder authorises the merchant to deduct multiple payments over an agreed period of time for a single purchase **RECURRING** - indicates a recurring transaction where the cardholder authorises the merchant to automatically debit their accounts for bill or invoice payments. This value only indicates to the acquirer that this is a recurring type payment; it does not mean that the merchant can use the Payment Server's Recurring Payment functionality. Note:This can only be used if the merchant has their privilege set to use this command, otherwise the transaction will be set to the merchant's default transaction source as defined by your Payment Provider.			
Optional	Alphanumeric	0,12	**SINGLE**

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Internet Transaction Data (ITD) Fields

The Internet Transaction Data (ITD) is a security feature plus adding functionality used for card not present transactions that add Internet data about this transaction. ITD data cannot be used at the same time as Airline Passenger Data (APD).

ITD or APD should only be implemented in 2-Party transactions as the payload is too large for a 3-Party style of transaction and the cardholder's browser will not perform a browser redirect.

For Card Not Present web merchants the ITD data need not contain fields:vpc_ITD_CustomerANI and vpc_ITD_CustomerANICallType.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Internet Transaction Data (ITD) Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_ITD_CustomerHostName			
Name of the server that the customer is connected to.			
Optional	Alphanumeric	0,60	phx.ow.aol.com
vpc_ITD_CustomerEmail			
Customer’s e-mail address			
Optional	Alphanumeric	0,60	jsmith@emailaddress.com
vpc_CustomerIpAddress			
Customer's Internet IP address - format: nnn.nnn.nnn.nnn			
Optional	Alphanumeric	15	127.142.005.056
vpc_ITD_CustomerBrowser			
Customer's HTTP browser type.			
Optional	Alphanumeric	0,60	MOZILLA/4.0 (COMPATIBLE; MSIE 5.0; WINDOWS 95)

vpc_ITD_MerchantSKU			
Unique SKU (Stock Keeping Unit) inventory reference number of product associated with this authorization request. For multiple items, enter the SKU for the single, most expensive item.			
Optional	Alphanumeric	0,15	TKDC315U

vpc_ITD_ShipMethodCode			

Shipping Method Code			
Optional	Alphanumeric	2,2	01 = Same Day 02 = Overnight / Next Day 03 = Priority, 2-3 Days 04 = Ground, 4 or more days 05 = Electronic Delivery
vpc_ITD_ShipToCountryCode			
Three-byte, numeric country code, example for USA: 840			
Optional	Numeric	0,3	840
vpc_ITD_CustomerANI			
ANI (Automatic Number Identification) specified phone number that the customer used to place order with merchant.			
Optional	Alphanumeric	10	6025551212
vpc_ITD_CustomerANICallType			
Telephone company-provided ANI ii (Information Identifier) coding digits associated with CUSTOMER ANI phone number that correspond to call-type; for example, cellular, government institution, etc.			
Optional	Alphanumeric	2	00

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Manual Authorisation ID

This field is given to selected Amex merchants that can be included with the transaction that gives them special privileges. This code is supplied by American Express to specific merchants and can be used in a number of transactions.

The presence of this code can allow a merchant to include Corporate Credit Card Level 3 (CPC3) data in a Purchase transaction.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Manual Auth ID			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_ManualAuthID			
An alphanumeric code of up to six characters used to specify the manual authorisation code supplied by the card issuer for the transaction.			
Optional	Alphanumeric	0,6	AB3456

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Airline Ticket Number Field

Although the ticket number was originally designed for the travel industry, Ticket Number functionality allows the merchant to enter any alphanumeric information to be stored on the Payment Server for that transaction.
Ticket number is passed with the Transaction Request and stored on the Payment Server. The ticket number is returned in the Transaction Response and is passed to the financial institution as part of certain transactions.

You can view the Ticket Number field in the search results of a Transaction Search using the Merchant Administration portal on the Payment Server.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

Airline Ticket Number Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_TicketNo			
The airline ticket number that is passed with the Transaction Request and stored on the Payment Server.			
Optional	Alphanumeric	0,15	A234567F

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Debit Card Fields

The debit card fields are applicable to debit card types such as Maestro or Solo. The debit card functionality allows the merchant to enter the type of account, card issue number, card start date, in the Transaction Request to be stored on the Payment Server for that transaction.

Note: Applies to 2-Party transactions and 3-Party with card details transactions.

Transaction Request Input Fields

Debit Card Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

BankAccountType			
The type of bank account the cardholder wants to use for the transaction. For example, Savings or Cheque. Valid values for this field are: CHQ — specifies that the cardholder wants to use the Cheque account linked to the card. SAV — specifies that the cardholder wants to use the Savings account linked to the card.			
Usage Notes: This identifier is mandatory if the card type is Maestro or Solo, and will be displayed in the Order Search results in the Merchant Administration portal on the Payment Server.			
Optional	Alphanumeric	3	SAV
vpc_CardIssueNumber			
The issue number of the card used with cards such as Maestro and Solo.			
Usage Notes: This identifier is mandatory if the card type is Maestro or Solo.			
Optional	Numeric	0,2	01
vpc_CardStartDate			
The start date of the card in yymm format used with cards such as Maestro and Solo. The value must be expressed as a 4-digit number (integer) with no white spaces or formatting characters. For example, an expiry date of May 2013 is represented as 1305.			
Usage Notes: This identifier is mandatory if the card type is Maestro or Solo.			
Optional	Numeric	4	1305

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Referral Message Fields

This response message occurs when the Acquirer needs to manually authorise the cardholder (by having the merchant contact them) as indicated by a **vpc_TxnResponseCode** '**E**'. See ***Transaction Response Codes*** on page 119.

The Authorisation code the merchant is given on contacting the Payment Provider is input using a '**Referral Transaction** on page 49'.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

There are no supplementary input fields in the Transaction Request for this Transaction Request.

Transaction Response Output Fields

Referral Message Output Field			
In addition to the standard output fields, the following field is also returned in the Transaction Response for both 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_AcquirerResponseAdvice			
Referral Message: This field is only present if vpc_TxnResponseCode is '**E**'. See Response Codes. This field is the referral message from the issuer. It may contain contact details to allow the merchant to contact the issuer directly to seek authorisation for the transaction. If Authorised the card company will provide a Manual Auth ID code that is input into the payment system using a '**Referral Transaction**'.			
Output	Alphanumeric	0,70	Please call John Doe at BankXYZ on 18004159896

Referral Processing Transaction Fields

Referral processing allows you to resubmit a referred initial transaction (Authorisation or Purchase transaction that received a "Refer to Issuer" acquirer response) as a new Authorisation or Purchase transaction with an authorisation code obtained from the issuer.

The card holder may be required to provide additional information in order for the issuer to approve the transaction and provide an authorisation code/Manual Auth ID.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Referral Processing Input Fields			
The following data fields must be included in a Transaction Request when performing a Referral transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_VirtualPaymentClientURL			
A fully qualified URL (starting with HTTPS://). It must be included in the merchant's application code to send transaction information to the Virtual Payment Client. https://<YOUR_VPC_URL>/vpcdps Note: This URL is supplied by the Payment Provider.			
Required	Alphanumeric	1,255	https://<YOUR_VPC_URL>/vpcdps

vpc_Version			
The version of the Virtual Payment Client API being used. The current version is 1.			
Required	Alphanumeric	1,8	1

vpc_Command			
Indicates the transaction type. This must be equal to '**doRequest**' for this type of transaction.			
Required	Alphanumeric	1,16	doRequest
vpc_RequestType			
This field is associated when the **vpc_Command** field equals '**doRequest**'. This must be equal to '**PAYMENT**' for this type of transaction.			
Required	Alphanumeric	1,20	PAYMENT

vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. Applicable values can be obtained from your Payment Services Provider. The value must be equal to '**doAuthorisedTransaction**' for this type of transaction.			
Required	Alphanumeric	1,25	doAuthorisedTransaction

vpc_AccessCode			
Authenticates the merchant on the Payment Server. This means that a merchant cannot access another merchant's Merchant Id. The access code is provided when the merchant profile is registered with a Payment Provider.			
Required	Alphanumeric	8	6AQ89F3

vpc_MerchTxnRef			
A unique value created by the merchant. **Usage Notes:** The Merchant Transaction Reference is used as a reference key to the Payment Server database to obtain a copy of lost/missing transaction receipts using the QueryDR function. It can also be used to identify a duplicate transaction if it is always kept unique for each transaction attempt. It can contain similar information to the vpc_OrderInfo field, but it must be unique for each transaction attempt if it is to be used properly. Typically, the vpc_MerchTxnRef is based on an order number, invoice number, timestamp, etc., but it should also reflect the transaction attempt. For example, if a cardholder has insufficient funds on their card and they are allowed to repeat the transaction with another credit card, the value may be INV1234/1 on the first attempt, INV1234/2 on the second attempt, and INV1234/3 on the third attempt. This identifier will be displayed in the Transaction Search results and also in the Download file (from Financial Transactions Search or Download Search Results link in Financial Transaction List) in the Merchant Administration portal on the Payment Server.			
Note: If "Enforce Unique Merchant Transaction Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's transactions.			
Required	Alphanumeric	1,40	ORDER958743-1

vpc_Merchant			
The unique Merchant Id assigned to a merchant by the Payment Provider. The Merchant ID identifies the merchant account against which settlements will be made.			
Required	Alphanumeric	1,16	TESTMERCHANT01
vpc_TransNo			
The unique Payment Server OrderID (Shopping Transaction) number of the existing order that has the referred transaction against it.			
Required	Numeric	1,19	10712

vpc_ManualAuthID			
An alphanumeric code of up to six characters used to specify the manual authorisation code supplied by the card issuer for the transaction.			
Optional	Alphanumeric	0,6	AB3456

Transaction Response Output Fields

There are no supplementary output fields in the Transaction Response for this Transaction Response.

Risk Management Fields

Risk Management is a security feature used for Card-Not-Present (CNP) transactions, which enables MSOs and merchants to mitigate fraud effectively using a set of business risk rules. These risk rules are configured to identify transactions of high/low risk thereby enabling merchants to accept, reject, or mark transactions for review based on risk assessment. For more information on the MSO and merchant rules, see Virtual Payment Client Integration Guide.

This feature is available for both 2-Party and 3-Party transactions. Though risk rules can be configured only through the Merchant Administration or Merchant Manager portal, transactions processed through the Virtual Payment Client will be assessed for risk, and the overall risk result for each authorisation and purchase will be returned in the Transaction Response. However, merchants using the Virtual Payment Client will not be able to make a review decision on the order — orders can be reviewed for processing or cancellation only through the Merchant Administration portal. You can view the overall risk result details in the search results of an Order Search using the Merchant Administration or Merchant Manager portal on the Payment Server.

Note: Risk Management is applicable only to:

- Merchants who have *May Use Risk Management* privilege enabled.

- Transaction modes, *Auth Then Capture* and *Purchase.* Standalone Captures, Standalone Refunds, etc., will not be assessed for risk.

The Risk Management feature includes the following fields:

- Bypass Risk Management — allows the merchant to process orders without performing risk checks and assessment of orders. The Bypass Risk Management field is passed with the Transaction Request and stored by the Payment Server. To transact using this field, the merchant operator must have *May Bypass Risk Management* privilege.

  Note: You cannot bypass MSO level risk rules.

- IP Address — allows the merchant to include the IP address of the cardholder in the Transaction Request — IP addresses are useful in identifying the location of the cardholder. The IP Address field is passed with the Transaction Request and stored by the Payment Server.
- Overall Risk Result — indicates the overall result of risk assessment for every authorisation or purchase, which is returned in the Transaction Response.
- Transaction Reversal Result — indicates the result of order reversal for each authorisation or purchase that occured due to risk assessment.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

Risk Management Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

RiskBypass			
Specifies whether the merchant wants to bypass risk checks and assessments for an order. Valid values for this field are: Y - indicates that the merchant wants to bypass risk checks. N - indicates that the merchant wants to perform risk checks and assessment on orders. This is the default value.			
Optional	Alphanumeric	1	Y
vpc_CustomerIpAddress			
Customer's Internet IP address - format: nnn.nnn.nnn.nnn			
Optional	Alphanumeric	15	127.142.005.056

Transaction Response Output Fields

Risk Management Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_RiskOverallResult			
The overall result of risk assessment for each authorisation or purchase. Valid values for this field are: ACC (Accept) — indicates that the order is accepted. REJ (Reject) — indicates that the order is rejected. REV (Review) — indicates that the order is marked for review. NCK (Not Checked) — indicates that the order is processed using the *Bypass Risk Management* option. It also implies a condition where neither MSO nor merchant risk rules are configured in the system. SRJ (System Reject) — indicates that the order is rejected at the system (MSO) level.			
Output	Alphanumeric	3	ACC
vpc_TxnReversalResult			
The result of order reversal for each authorisation or purchase that occured due to risk assessment. Orders rejected after the financial transaction due to risk assessment are automatically reversed by the system. Valid values for this field are: OK — indicates that the order was reversed successfully. FAIL — indicates that the attempt to reverse the order failed. NA (Not Supported) — indicates that the acquirer does not support reversal of the required transaction so the reversal failed.			
Output	Alphanumeric	4	OK

Dynamic Currency Conversion (DCC) Fields

Dynamic Currency Conversion (DCC) is a feature that allows merchants receiving payments in foreign (target) currencies to perform transactions in the base(merchant-configured) currency. The DCC server calculates the exchange rates for foreign currencies before performing the transaction.

Transaction Request Input Fields

Dynamic Currency Conversion Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Dcc_OfferState			
Indicates currency-conversion acceptance details. Valid values are: ▪ ACC - DCC offered and accepted ▪ DEC - DCC offered but declined ▪ NOF - DCC not offered for specific request ▪ NAT - DCC was not attempted			
Optional	Alpha	3	ACC
vpc_Dcc_BaseAmount			
The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: This field is mandatory if vpc_Dcc_OfferState is set to "ACC".			
Optional	Numeric	1,12	1250
vpc_Dcc_BaseCurrency			
The base currency of the order expressed as an ISO 4217 alphanumeric code.			
Note: This field is mandatory if vpc_Dcc_OfferState is set to "ACC".			
Optional	Alpha	3	USD
vpc_Dcc_ExchangeRate			
The currency-conversion exchange rate used to determine the vpc_Currency, expressed to four decimal places.			
Note: This field is mandatory if vpc_Dcc_OfferState is set to "ACC".			

Optional	Numeric	13.4	42.4678

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Custom Payment Plans

Custom Payment Plan is an installment payment option configured by the merchant for the cardholder using the plan types offered by the MSO. Plan types are MSO-dependent installment payment options which are determined by the merchant and MSO based on the cardholder's requirements. The payment plans enable cardholders to break the purchase order into a number of monthly installments or defer payments for purchases and then pay using monthly installments. Generally there is a maximum of 99 installments and/or deferral months.

Note: Applicable only to transactions using Mexican Peso currency.

A custom payment plan typically includes:

- Plan Name

  A merchant-supplied identifier for the payment plan. The Payment Plan Name is unique per Payment Plan Type for the merchant.

- Plan ID

  A auto-generated unique identifier for the payment plan. The Plan ID is unique across all Payment Plan Types for the merchant

- Installment Months (if applicable to the plan type)

  The installment terms in months configured for the payment plan.

- Deferral Months (if applicable to the plan type)

  The deferral terms in months configured for the payment plan.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

Custom Payment Plan Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_CustomPaymentPlanPlanId			
A auto-generated unique identifier for the payment plan. The Plan ID is unique across all Payment Plan Types for the merchant.			
Required	Alphanumeric	1,16	BPWOI1
vpc_NumPayments			
The number of monthly installments. Numeric values 1 to 99, which is the maximum value.			
Optional	Integer	1,99	10
vpc_NumDeferrals			
The number of deferrals in months. Numeric values 1 to 99, which is the maximum value.			

Optional	Integer	1,99	3

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Plan 'N' Fields

Plan 'N' is a financial payment option available in some countries. Plan N, allows cardholders to defer payments for purchases from an eligible merchant into monthly installments. The merchant determines the number of installments and accepts the applicable charges and payment plan conditions with American Express. The card member is billed in installments without any interest while the merchant is paid in the agreed payment plan less the applicable charges.

Generally there is a maximum of 24 installments.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

Plan 'N' Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_PaymentPlan			
Indicates the type of instalment plan. This must be equal to '**PlanN**' for a Plan 'N' payment transaction.			
Optional	Alpha	8	PlanN

vpc_NumPayments			
The number of payments. Numeric values 1 to 24, which is the maximum value.			
Optional	Integer	1,2	10

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

Plan 'Amex' Fields

Plan 'Amex' or Deferred Payment Plan (DPP) is a payment method available to cardholders in some markets. In Plan 'Amex', the merchant is paid in full less applicable discount rate and the cardholder is billed in installments plus the applicable interest rate.

Plan 'Amex' transactions in Brazil are done as an initial inquiry followed by an authorization. This is to satisfy statutory requirements in that country to provide information about the amount of interest charged to the customer.

Note: Applies to 2-Party and 3-Party transactions.

Transaction Request Input Fields

Plan 'Amex' Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_PaymentPlan			
Indicates the type of instalment plan. This must be equal to '**PlanAmex**' for a Plan 'Amex' payment transaction.			
Optional	Alpha	8	PlanAmex

vpc_NumPayments			
The number of payments. Numeric values 1 to 24, which is the maximum value.			
Optional	Integer	1,2	10

Transaction Response Output Fields

Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_SaleAmount
This is equal to the transaction vpc_Amount.

Input	Numeric	0,10	100000

vpc_InterestRate			
The interest rate charged by American Express. This may differ from the interest rate being charged by the merchant.			
Output	Alphanumeric	0,10	7

Plan 'Amex' Inquiry Fields

Plan 'Amex' or Deferred Payment Plan (DPP) is a payment method available to cardholders in some markets. In Plan 'Amex', the merchant is paid in full less applicable discount rate and the cardholder is billed in installments plus the applicable interest rate.

Plan 'Amex' transactions in Brazil are done as an initial inquiry followed by an authorization. This is to satisfy statutory requirements in that country to provide information about the amount of interest charged to the customer.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Plan 'Amex' Inquiry Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_PaymentPlan			
Indicates a flag in the digital order that specifies that this transaction is an inquiry transaction. The value for this field must be equal to 'inquiryTransaction' for a Plan 'Amex' Inquiry transaction. An inquiry transaction will not process a payment but returns payment plan options allowing the cardholder to choose the appropriate plan. **Warning: To process a PlanAmex transaction, (not an inquiry transaction), do not include this field in the Transaction Request.**			
Optional	Alpha	20	inquiryTransaction

vpc_NumPayments			
The number of payments. Numeric values 1 to 24, which is the maximum value.			
Optional	Integer	1,2	10

Transaction Response Output Fields

Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_NumOccurrences			
The number of payment plan options.			
Output	Numeric	0,10	4
vpc_FinalAmountX			
This is equal to the total amount being repaid to Amex, including interest for this plan. X is an integer that varies between 1 and the number of payment options (maximum of 4 options) available.			
Output	Numeric	0,10	1001587
vpc_NumPaymentsX			
The number of payments. This has a maximum value of '24' for this plan. X is an integer that varies between 1 and the number of payment options (maximum of 4 options) available.			
Output	Numeric	0,2	9
vpc_PaymentAmountX			
The value of each instalment for this plan. X is an integer that varies between 1 and the number of payment options (maximum of 4 options) available.			
Output	Numeric	0,10	10051

Prior Authorisation Fields

The **Prior Authorised Transaction** command allows a merchant to resubmit a transaction with an authorisation code obtained from the Issuer. However, the amount cannot be altered in this transaction.

This transaction may or may not be followed by a referred initial transaction. Referral Message.

Note: Applies to 2-Party transactions.

Transaction Request Input Fields

Prior Authorisation Input Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**doRequest**' for this type of transaction.			
Required	Alphanumeric	1,16	doRequest

vpc_RequestType			
This field is associated when the **vpc_Command** field equals '**doRequest**'. This must be equal to '**PAYMENT**' for this type of transaction.			
Required	Alphanumeric	1,20	PAYMENT

vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. The value must be equal to '**doAuthorisedTransaction**' for this type of transaction.			
Required	Alphanumeric	1,20	doAuthorisedTransaction

vpc_TransNo			
This is the unique Payment Server Order Number (Shopping Transaction Number) generated by the Payment Server for the initial transaction.			
Required	Numeric	1,19	10712

vpc_ManualAuthID			
An alphanumeric code of up to six characters used to specify the manual authorisation code supplied by the card issuer for the transaction.			
Optional	Alphanumeric	0,6	AB3456

Transaction Response Output Fields

The Transaction Response fields returned for this option are the same as those returned for the standard 2-Party Transaction output fields.

Payment Authentication

Payment Authentications are designed to prevent credit card fraud by authenticating cardholders when performing transactions over the Internet by using the 3-Domain Secure™ (3-D Secure or 3DS) protocol developed by Visa.

A 3-D Secure transaction is performed immediately before a merchant performs a payment transaction, that is, an Authorisation transaction in the Auth/Capture mode, or a Purchase transaction in the Purchase mode. Authentication ensures that the card is being used by its legitimate owner.

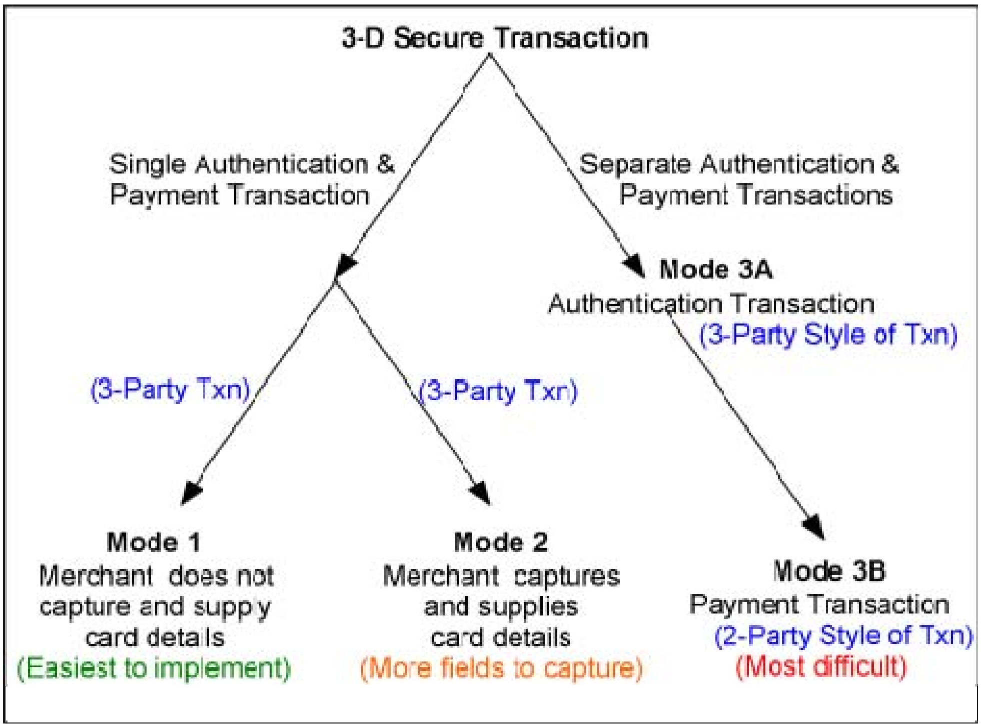
During a transaction, 3DS authentication allows the merchant to authenticate the cardholder by redirecting them to their card issuer where they enter a previously registered password.

Merchants using 3DS can be configured to block any transaction that fails 3DS authentication. A transaction is considered to fail 3DS authentication if it results in a Verification Security Level of '07'. A blocked transaction results in a Dialect Response Code of 'B', which is included in the DR and displayed in the Financial Transaction Details page.

Note: 3DS Authentication can only take place if the merchant is using a 3-Party model of transaction as the cardholder's browser has to be redirected to their card issuing bank where they enter their secret password. This is performed by the Payment Server if the cardholder is enrolled in the 3DS schemes.

Payment Authentication 3-D Secure transaction modes

The following diagram shows an overview of the Payment Authentication 3-D Secure transaction modes.


3-D Secure Transaction
Single Authentication & Payment Transaction
Separate Authentication & Payment Transactions
Mode 3A
Authentication Transaction
(3-Party Style of Txn)
(3-Party Txn)
(3-Party Txn)
Mode 1
Merchant does not capture and supply card details
(Easiest to implement)
Mode 2
Merchant captures and supplies card details
(More fields to capture)
Mode 3B
Payment Transaction
(2-Party Style of Txn)
(Most difficult)

The available 3-D Secure transaction modes are:

1. **Mode 1 - Combined 3-Party Authentication and Payment transaction** - the merchant uses the Payment Server to perform the authentication and payment in the one transaction.

   The *Payment Server collects the cardholder's card details* and not the merchant's application. The Payment Server redirects the cardholder to the card-issuing institution to enter their 3-D Secure password. If the Authentication is performed correctly the Payment Server uses the Authentication information to perform the payment transaction.

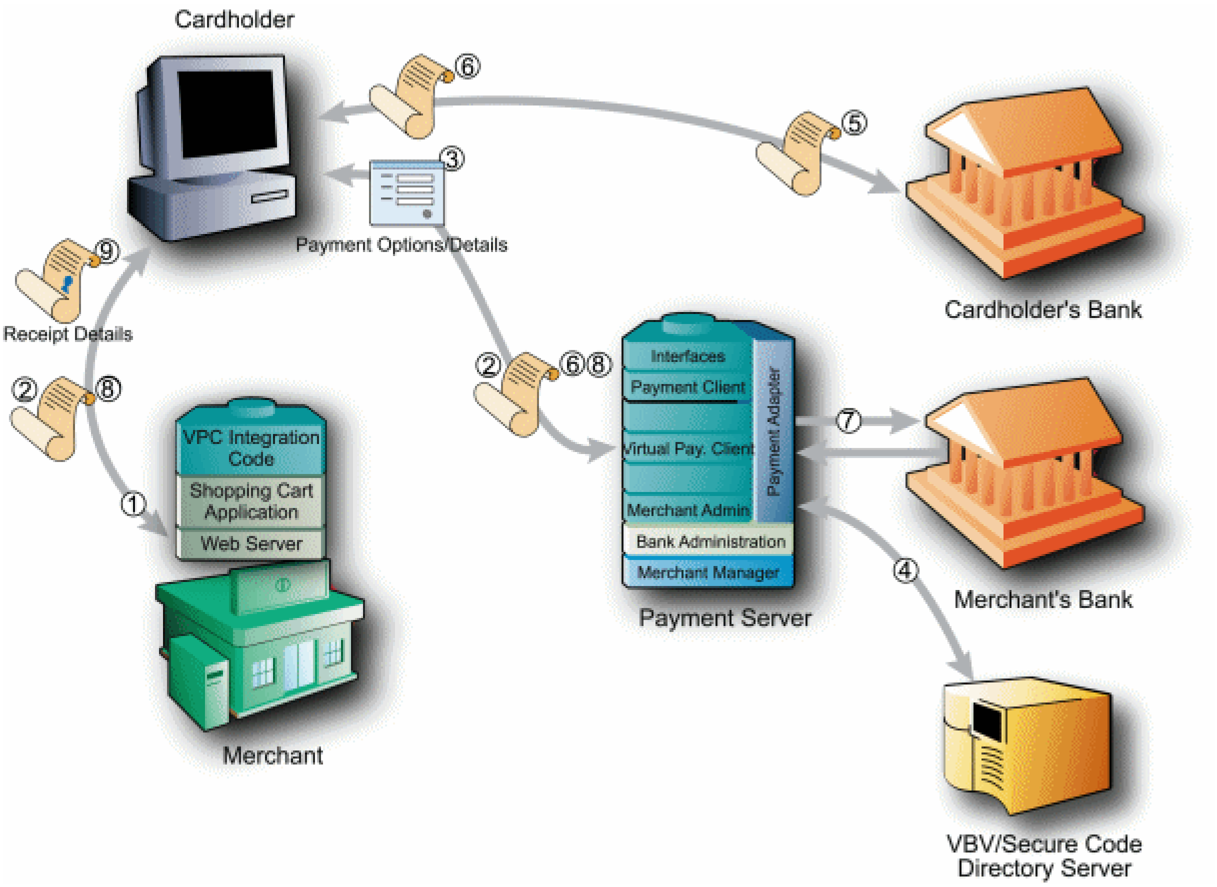
2. **Mode 2 - Combined 3-Party Authentication and Payment transaction, (merchant collects card details)** - the merchant uses the Payment Server to perform the authentication and payment in the one transaction.

   The *merchant's application collects the cardholder's card details* and sends them to the Payment Server, which redirects the cardholder to the card-issuing institution to enter their 3-D Secure password. If the Authentication is performed correctly the Payment Server uses the Authentication information to perform the payment transaction.

3. **Mode 3a - 3-Party - Authentication Only transaction** - the merchant uses the Payment Server to perform an authentication transaction and the payment transaction is processed as a separate transaction. This gives the merchant complete control as to when and if a payment transaction should proceed. The Authentication operation outputs become the inputs for a 3-Party with card details transaction. The merchant needs to collect card details.

   Mode 3b - 2-Party Style Pre-Authenticated Payment transaction - the merchant may use the 3-Party - Authentication only transaction through the Payment Server or an external authentication provider to perform the 3-D Secure Authentication, and use the outputs from this operation to perform a 2-Party payment transaction through the Payment Server. The merchant needs to collect card details.

Information Flow of a 3D-Secure Authentication/Payment transaction



If you have been enabled to use 3-D Secure, the information flow for 3-D Secure where the Payment Server collects the card details (Mode1) is as follows:

1 A cardholder browses the application, selects a product and enters their shipping details into the merchant's application at the checkout page.

2 The cardholder clicks a pay button and your application sends the payment Transaction Request to the Payment Server by redirecting the cardholder's Internet browser to the Payment Server.

3 The Payment Server prompts the cardholder for the card details.

4 If the card is a Visa or MasterCard, for example, the Payment Server then checks with the VBV or SecureCode Directory Server to determine if the card is enrolled in either the Verified by Visa™ (Visa 3-Domain Secure) or MasterCard SecureCode™ (MasterCard 3-Domain Secure) scheme.
If the card is not enrolled in payment authentication scheme then go to Step 7.
If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuing site for authentication.

5 If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuer's site for authentication. The card issuer's server displays the cardholder's secret message and the cardholder enters their secret password, which is checked against the Issuing bank's database.

6 At the completion of the authentication stage, the cardholder is redirected back to the Payment Server indicating whether or not the cardholder's password matched the password in the database.

If the cardholder was not authenticated correctly, then the payment does not take place and the cardholder is redirected back to the merchant's site with a Transaction Response containing details to indicate the authentication failed - see step 8.

7 If the cardholder was authenticated correctly, or Payment Authentication did not occur the Payment Server continues with processing the transaction with the results of the authentication attempt.

8 The Payment Server then redirects the cardholder back to merchant's site with the Transaction Response. The Transaction Response contains the result of the transaction.

9 The application processes the Transaction Response and displays the receipt.

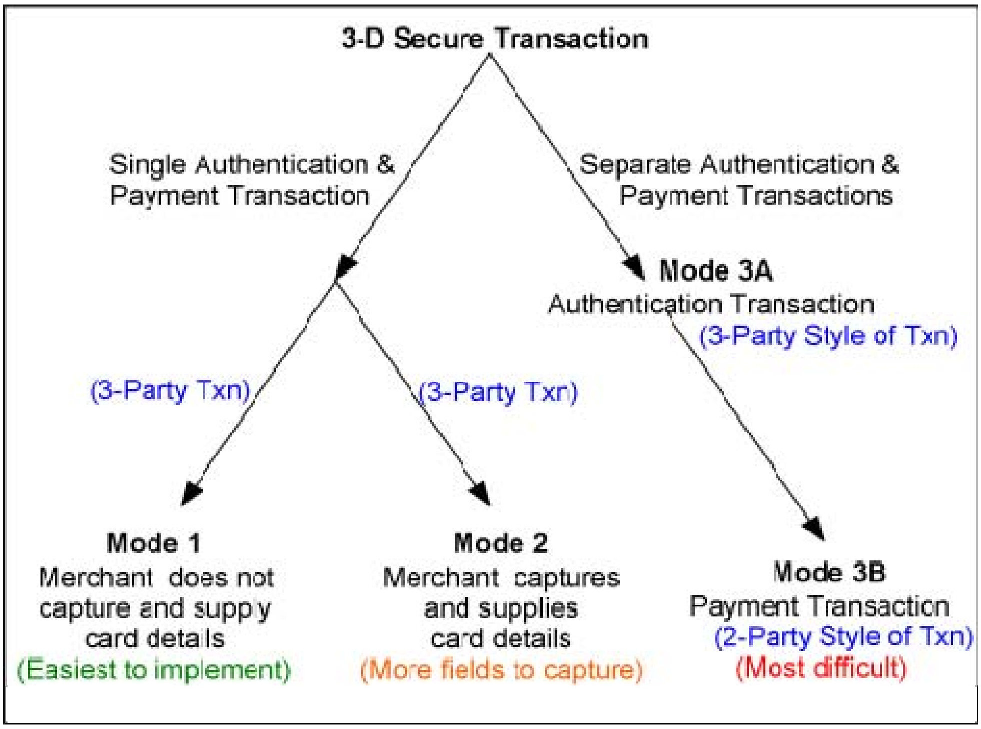
Note: If the cardholder is enrolled in the 3D Secure scheme but is not authenticated correctly, for example, because the cardholder may have entered their password incorrectly 3 times, then the merchant's application is sent a **vpc_TxnResponseCode** code of '**F**' to indicate the cardholder failed the authentication process and the transaction does not proceed.

Mode 2 and Mode 3a are slight variations on the above information flow. In mode 2 and mode 3a the merchant collects the card details and passes them through, which means step 3 is eliminated.

For Mode 3a step 7 is also eliminated, the payment being performed through a separate 2-Party transaction after the Authentication.

Payment Authentication 3-D Secure transaction modes

The following diagram shows an overview of the Payment Authentication 3-D Secure transaction modes.


3-D Secure Transaction
Single Authentication & Payment Transaction
Separate Authentication & Payment Transactions
Mode 3A
Authentication Transaction
(3-Party Style of Txn)
(3-Party Txn)
(3-Party Txn)
Mode 1
Merchant does not capture and supply card details
(Easiest to implement)
Mode 2
Merchant captures and supplies card details
(More fields to capture)
Mode 3B
Payment Transaction
(2-Party Style of Txn)
(Most difficult)

1 **Mode 1 - Combined 3-Party Authentication and Payment transaction** - the merchant uses the Payment Server to perform the authentication and payment in one transaction.

   The *Payment Server collects the cardholder's card details* and not the merchant's application. The Payment Server redirects the cardholder to the card-issuing bank to enter their 3-D Secure password. If the Authentication is performed correctly the Payment Server uses the Authentication information to perform the payment transaction.

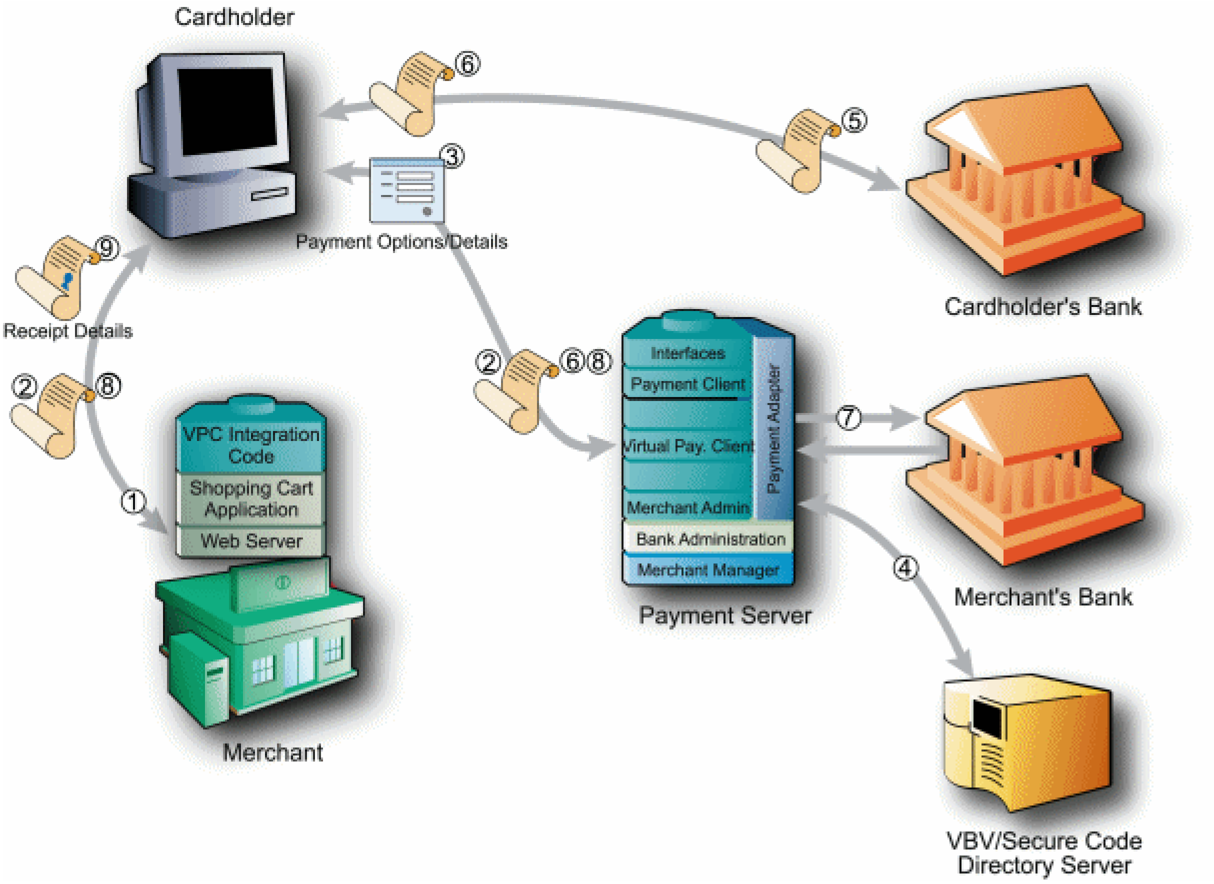
2 **Mode 2 - Combined 3-Party Authentication and Payment transaction, (merchant collects card details)** the merchant uses the Payment Server to perform the authentication and payment in the one transaction.

   The *merchant's application collects the cardholder's card details* and sends them to the Payment Server, which redirects the cardholder to the card-issuing bank to enter their 3-D Secure password. If the Authentication is performed correctly the Payment Server uses the Authentication information to perform the payment transaction.

3 **Mode 3a - 3-Party - Authentication Only transaction** - the merchant uses the Payment Server to perform an authentication transaction and the payment transaction is processed as a separate transaction. This gives the merchant complete control as to when and if a payment transaction should proceed. The *merchant's application collects the cardholder's card details* and sends them to the Payment Server, which redirects the cardholder to the card-issuing bank to enter their 3-D Secure password.
   The Authentication operation outputs become the inputs for a 3-Party with card details transaction.

   Mode 3b - 2-Party Style Pre-Authenticated Payment transaction - the merchant may use the 3-Party - Authentication only transaction through the Payment Server or an external authentication provider to perform the 3-D Secure Authentication, and use the outputs from this operation to perform a 2-Party payment transaction through the Payment Server.

Information Flow of a 3D-Secure Authentication/Payment transaction



If you have been enabled to use 3-D Secure, the information flow for 3-D Secure where the Payment Server collects the card details (Mode1) is as follows:

1 A cardholder browses the application, selects a product and enters their shipping details into the merchant's application at the checkout page.

2 The cardholder clicks a pay button and your application sends the payment Transaction Request to the Payment Server by redirecting the cardholder's Internet browser to the Payment Server.

3 The Payment Server prompts the cardholder for the card details.

4 If the card is a Visa or MasterCard, for example, the Payment Server then checks with the VBV or SecureCode Directory Server to determine if the card is enrolled in either the Verified by Visa™ (Visa 3-Domain Secure) or MasterCard SecureCode™ (MasterCard 3-Domain Secure) scheme.
If the card is not enrolled in payment authentication scheme then go to Step 7.
If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuing site for authentication.

5 If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuer's site for authentication. The card issuer's server displays the cardholder's secret message and the cardholder enters their secret password, which is checked against the Issuing bank's database.

6 At the completion of the authentication stage, the cardholder is redirected back to the Payment Server indicating whether or not the cardholder's password matched the password in the database.

If the cardholder was not authenticated correctly, then the payment does not take place and the cardholder is redirected back to the merchant's site with a Transaction Response containing details to indicate the authentication failed - see step 8.

7 If the cardholder was authenticated correctly, or Payment Authentication did not occur the Payment Server continues with processing the transaction with the results of the authentication attempt.

8 The Payment Server then redirects the cardholder back to merchant's site with the Transaction Response. The Transaction Response contains the result of the transaction.

9 The application processes the Transaction Response and displays the receipt.

Note: If the cardholder is enrolled in the 3D Secure scheme but is not authenticated correctly, for example, because the cardholder may have entered their password incorrectly 3 times, then the merchant's application is sent a **vpc_TxnResponseCode** code of '**F**' to indicate the cardholder failed the authentication process and the transaction does not proceed.

Mode 2 and Mode 3a are slight variations on the above information flow. In mode 2 and mode 3a the merchant collects the card details and passes them through, which means step 3 is eliminated.

For Mode 3a step 7 is also eliminated, the payment being performed through a separate 2-Party transaction after the Authentication.

Advantages and Disadvantages of the 3-D Secure modes of transaction

Mode	Advantages	Disadvantages
Mode 1 3 Party Authentication and Payment transaction mode	▪ Simple to implement. ▪ The Payment Provider collects the cardholder's card details and not the merchant, which provides highest level of security for the cardholder's card details.	▪ The merchant is not able to use their own branding throughout the whole transaction, as the Payment Provider displays their own branding while the card details are being captured. ▪ If the cardholder is not enrolled in 3-D Secure, or the authentication could not be performed, the authentication will not take place and the transaction will automatically move into the payment stage.
Mode 2 3 Party Authentication and Payment transaction (Merchant collects card details)	▪ Suits a merchant that normally collects all the card details. ▪ Branding of the payment pages on the website remains consistent throughout the whole transaction, except the screen where the cardholder enters their password for 3-D secure.	▪ If the cardholder is not enrolled in 3-D Secure the authentication will not take place and the transaction will automatically move into the payment stage.
Mode 3a 3 Party Authentication Only transaction mode	▪ Suits a merchant that normally collects all the card details. ▪ Branding of the payment pages on the website remains consistent throughout the whole transaction, except the screen where the cardholder enters their password for 3-D secure.	▪ It consists of two separate transactions, the Authentication and the Payment, which can be more difficult for a merchant to integrate.
Mode 3b 2 Party Pre-Authenticated transaction mode	▪ Gives the merchant maximum control of the transaction. If the cardholder is not enrolled in 3-D Secure, then the merchant's application can stop the transaction from progressing to the Payment stage providing full control over the transaction risk. ▪ Branding remains consistent throughout the whole transaction, except for the one screen where the cardholder enters their 3-D Secure password.	▪ Can only be performed if the merchant collects all the card details. ▪ It consists of two separate transactions, the Authentication and the Payment, which can be more difficult for a merchant to integrate.

Mode 1 - 3-Party Authentication & Payment Transaction: (Payment Server collects card details)

The 3-Party Authentication and Payment transaction mode uses the basic 3-Party style of transaction.

Mode 1 Transaction Request Input Fields

There are no additional input fields in the Transaction Request to add 3-D Secure authentication to a standard 3-Party transaction.

Mode 1 Transaction Response Outputs

The outputs from this transaction type are the same as ***Mode 2 type transactions*** on page 72.

Mode 2 - 3-Party Authentication & Payment Txn: (Merchant collects card details)

If you want to keep branding consistent throughout the transaction you can pass in extra fields to a 3-Party transaction, but you do need your Payment Provider to enable you to use card details in the Transaction Request. These fields are outlined below.

Mode 2 Transaction Request Input Fields

Card Details in Transaction Request Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Card			
Used in External Payment Selection to determine what type of card is used. The field is case sensitive, and must comply with each of the card types valid in the Payment Server. This varies from Payment Server to Payment Server. The possible values are shown in ***External Payment Selection (EPS)*** on page 123. To check the card types available for your Payment Provider, perform a 3-Party transaction and go to the Payment Server card selection page in a browser. Run the cursor over each card logo.The 'card' and 'gateway' values are displayed at the bottom of the browser window.			
Required	Alphanumeric	3,16	Visa

vpc_Gateway			
Determines the type of payment gateway functionality. The field is case sensitive, and must comply with the gateways that are valid in the Payment Server. Valid values for this field are: ▪ **ssl** — specifies the gateway for all standard 3-Party transactions ▪ **threeDSecure** — specifies the gateway for a 3-D Secure Mode 3a-3-party Style Authentication Only transaction. Note: For most transactions the value of this field will be '**ssl**'			
Required	Alphanumeric	3,15	ssl

vpc_CardNum			
The number of the card used for the transaction. The format of the Card Number is based on the Electronic Commerce Modeling Language (ECML) and, in particular, must not contain white space or formatting characters.			
Required	Numeric	15,19	5123456789012346

vpc_CardExp			
The expiry date of the card in the format YYMM. The value must be expressed as a 4-digit number (integer) with no white space or formatting characters. For example, an expiry date of May 2013 is represented as 1305.			
Required	Numeric	4	1305

vpc_CardSecurityCode			
The Card Security Code (CSC), also known as CVV(Visa), CVC2(MasterCard) or CID/4DBC(Amex) or CVV2, which is printed, not embossed on the card. It compares the code with the records held in the card issuing institution's database.			
Optional	Numeric	3,4	985

vpc_Desc			
An optional field that the merchant may supply in the Transaction Request as a description of the transaction. This description will be displayed on the Verified by Visa^{TM} page where the cardholder types in their secret password. Note: This is only used for Verified by Visa^{TM} transactions and cannot be used for MasterCard SecureCode^{TM} as this field is not displayed. The field can only be used if the merchant collects the card details and passes them in. If the Payment Server is used to collect the card details, the merchant cannot use the Desc field.			
Optional	Alphanumeric	0,125	This is some description about the Verified by Visa^{TM} transaction.

Mode 2 Transaction Response Output Fields

These fields are only returned in the Transaction Response if the transaction is a 3-D Secure payment authentication. Other cards like Bankcard and American Express will not return these additional fields. You must also be enabled on the Payment Server by your Payment Provider to perform 3-D Secure payment authentications.

The **vpc_TxnResponseCode** can be used to determine if the authentication passed or failed. If the **vpc_TxnResponseCode** is equal to '**F**', the Authentication process failed and no payment took place. If the **vpc_TxnResponseCode** is not equal to '**F**', the payment authentication process was attempted and the payment process takes place.

If a payment authentication has been successful, extra fields are returned in the Transaction Response for a 3-D Secure payment authentication. These fields are not used by you but are returned to allow you to store them as a record of authentication for the transaction, which can be used to resolve disputes. They cannot be used again for any future transactions.

All payment authentication transactions use a **vpc_VerStatus** response code value to show whether the card authentication was successful or not. For details of this code, please see ***3-D Secure Status Codes*** on page 128.

Mode 2 Payment Authentication Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for this 3-Party transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_3DSECI			
The 3-D Secure Electronic Commerce Indicator, which is set to '05' when the cardholder authenticates OK, and '07' when the cardholder is not enrolled. (These values may change depending on the locale or issuer).			
Output	Numeric	0,2	07

vpc_3DSXID			
It is a unique transaction identifier that is generated by the Payment Server on behalf of the merchant to identify the 3DS transaction. It is a 20-byte field that is Base64 encoded to produce a 28-character value.			
Output	Alphanumeric	0,28	uyPfGIgsoFQhklkIsto+IFWs92s=

vpc_3DSenrolled			
This field indicates if the card is within an enrolled range. This is the value of the VERes.enrolled field. It will take values (**Y** - Yes, **N** - No, **U** - Unavailable for Checking).			
Output	Alpha	1	N

vpc_3DSstatus			
This field is only included if payment authentication was attempted and a PARes was received by the MPI. It will take values (**Y** - Yes, **N** - No, **A** - Attempted Authentication, **U** - Unavailable for Checking).			
Output	Alpha	0,1	N

vpc_VerToken			
This value is generated by the card issuer as a token to prove that the cardholder authenticated OK. This is a base64 encoded value.			
Output	Alphanumeric	28	gIGCg4SFhoeIiYqLjI2Oj5CRkpM=

vpc_VerType			
This field will either be '**3DS**' 3-D Secure incorporating one of the 3-D Secure schemes. For example, Verified by Visa or MasterCard SecureCode or '**SPA**' - Secure Payment Authentication from MasterCard (rarely used).			
Output	Alphanumeric	0,3	3DS

vpc_VerSecurityLevel

The Verification Security Level is generated at the card issuer as a token to prove that the cardholder was enrolled and authenticated OK. It is shown for all transactions except those with authentication status “Failure”. This field contains the security level to be used in the AUTH message. MasterCard '**0**' - Merchant not participating (a merchant will not see this if they are configured for MasterCard SecureCode). MasterCard '**1**' - Cardholder not participating. MasterCard '**2**' - Cardholder authenticated. Visa '**05**' - Fully Authenticated. Visa '**06**' - Not authenticated (cardholder not participating). Visa '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid. American Express '**05**' - Fully Authenticated. American Express '**06**' - Not authenticated (cardholder not participating). American Express '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid.			
Output	Numeric	0,2	06

vpc_VerStatus			
The status codes used by the Payment Server to show whether the payment authentication was successful or not. ***3-D Secure Status Codes*** on page 128.			
Output	Alphanumeric	1	N

Mode 3a - 3-Party Style Authentication Only Transaction: (Merchant collects card details)

In certain cases a merchant may want to perform an Authentication of the cardholder separately to a payment transaction. This could because the merchant only wants to take a payment from cardholders that are both:

1 Enrolled in 3-D Secure **and:**

2 That cardholder is correctly Authenticated

In a normal operation, if the cardholder is not enrolled in 3-D Secure, the payment still goes ahead. In Mode 3a if the cardholder is not enrolled they are returned to the merchant site before the payment goes ahead.

The following fields are added to a standard 3-Party transaction to perform an Authentication Only transaction. **No payment is carried out with this transaction.** The merchant must have the EPS privilege, and cardholders enrolled. The merchant must be set up to provide the card details on the Transaction Request.

To perform a payment, the outputs from this transaction are fed as additional inputs to a standard 2-Party transaction.

Mode 3a Payment Authentication Only Input Fields

Payment Authentication Only Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Card			
Used in External Payment Selection to determine what type of card is used. The field is case sensitive, and must comply with each of the card types valid in the Payment Server. This varies from Payment Server to Payment Server. The possible values are shown in ***External Payment Selection (EPS)*** on page 123. To check the card types available for your Payment Provider, perform a 3-Party transaction and go to the Payment Server card selection page in a browser. Run the cursor over each card logo.The 'card' and 'gateway' values are displayed at the bottom of the browser window.			
Required	Alphanumeric	3,16	Visa

vpc_Gateway			
Determines the type of payment gateway functionality. The field is case sensitive, and must comply with the gateways that are valid in the Payment Server. Valid values are shown in the External Payment Selection (EPS) section. For an Authentication Only transaction the field value must be '**threeDSecure**'			
Required	Alphanumeric	3,15	threeDSecure

vpc_CardNum

The number of the card used for the transaction. The format of the Card Number is based on the Electronic Commerce Modeling Language (ECML) and, in particular, must not contain white space or formatting characters.			
Required	Numeric	15,19	5123456789012346

vpc_CardExp			
The expiry date of the card in the format YYMM. The value must be expressed as a 4-digit number (integer) with no white space or formatting characters. For example, an expiry date of May 2013 is represented as 1305.			
Required	Numeric	4	1305

vpc_Desc			
An optional field that the merchant may supply in the Transaction Request as a description of the transaction. This description will be displayed on the Verified by Visa^{TM} page where the cardholder types in their secret password. Note: This is only used for Verified by Visa^{TM} transactions and cannot be used for MasterCard SecureCode^{TM} as this field is not displayed. The field can only be used if the merchant collects the card details and passes them in. If the Payment Server is used to collect the card details, the merchant cannot use the Desc field.			
Optional	Alphanumeric	0,125	This is some description about the Verified by Visa^{TM} transaction.

Mode 3a Payment Authentication Only Output Fields

These fields are only returned in the Transaction Response if the transaction is a 3-D Secure payment authentication. You must be enabled on the Payment Server by your Payment Provider to perform 3-D Secure payment authentications.

The **vpc_TxnResponseCode** is used to determine if the authentication passed or a failed.

If the **vpc_TxnResponseCode** is not equal to '**F**', the payment authentication passed OK and the Authentication process has completed satisfactorily.

If the **vpc_TxnResponseCode** is equal to '**F**', the Authentication process failed and no payment will take place.

If a payment authentication has been successful, extra fields are returned in the Transaction Response for a 3-D Secure payment authentication. The fields are returned to be included in the mode 3b pre-authentication payment transaction.They cannot be used again for any future transactions.

All payment authentication transactions use a **vpc_VerStatus** response code value to show whether the card authentication was successful or not. For details of this code, please see ***3-D Secure Status Codes*** on page 128.

Payment Authentication Output Fields
In addition to the standard output fields, the following fields are also returned in the Transaction Response for both 2-Party and 3-Party transactions.
Field Name
Field Description

<table>
<tr><th>Returned Input or Output</th><th>Field Type</th><th>Min, Max or Set Field Length</th><th>Sample Data</th></tr>
<tr><td colspan="4">vpc_3DSECI</td></tr>
<tr><td colspan="4">The 3-D Secure Electronic Commerce Indicator, which is set to '05' when the cardholder authenticates OK, and '07' when the cardholder is not enrolled. (These values may change depending on the locale or issuer).</td></tr>
<tr><td>Output</td><td>Numeric</td><td>0,2</td><td>07</td></tr>
<tr><td colspan="4">vpc_3DSXID</td></tr>
<tr><td colspan="4">It is a unique transaction identifier that is generated by the Payment Server on behalf of the merchant to identify the 3DS transaction. It is a 20-byte field that is Base64 encoded to produce a 28-character value.</td></tr>
<tr><td>Output</td><td>Alphanumeric</td><td>0,28</td><td>uyPfGIgsoFQhklkIsto+IFWs92s=</td></tr>
<tr><td colspan="4">vpc_3DSenrolled</td></tr>
<tr><td colspan="4">This field indicates if the card is within an enrolled range. This is the value of the VERes.enrolled field. It will take values (<b>Y</b> - Yes, <b>N</b> - No, <b>U</b> - Unavailable for Checking).</td></tr>
<tr><td>Output</td><td>Alpha</td><td>1</td><td>N</td></tr>
<tr><td colspan="4">vpc_3DSstatus</td></tr>
<tr><td colspan="4">This field is only included if payment authentication was attempted and a PARes was received by the MPI. It will take values (<b>Y</b> - Yes, <b>N</b> - No, <b>A</b> - Attempted Authentication, <b>U</b> - Unavailable for Checking).</td></tr>
<tr><td>Output</td><td>Alpha</td><td>0,1</td><td>N</td></tr>
<tr><td colspan="4">vpc_VerToken</td></tr>
<tr><td colspan="4">This value is generated by the card issuer as a token to prove that the cardholder authenticated OK. This is a base64 encoded value.</td></tr>
<tr><td>Output</td><td>Alphanumeric</td><td>28</td><td>gIGCg4SFhoeIiYqLjI2Oj5CRkpM=</td></tr>
<tr><td colspan="4">vpc_VerType</td></tr>
<tr><td colspan="4">This field will either be '<b>3DS</b>' 3-D Secure incorporating one of the 3-D Secure schemes. For example, Verified by Visa or MasterCard SecureCode or '<b>SPA</b>' - Secure Payment Authentication from MasterCard (rarely used).</td></tr>
<tr><td>Output</td><td>Alphanumeric</td><td>0,3</td><td>3DS</td></tr>
<tr><td colspan="4">vpc_VerStatus</td></tr>
<tr><td colspan="4">The status codes used by the Payment Server to show whether the payment authentication was successful or not. <b><i>3-D Secure Status Codes</i></b> on page 128.</td></tr>
<tr><td>Output</td><td>Alphanumeric</td><td>1</td><td>N</td></tr>
<tr><td colspan="4">vpc_VerSecurityLevel</td></tr>
</table>

The Verification Security Level is generated at the card issuer as a token to prove that the cardholder was enrolled and authenticated OK. It is shown for all transactions except those with authentication status “Failure”. This field contains the security level to be used in the AUTH message. MasterCard '**0**' - Merchant not participating (a merchant will not see this if they are configured for MasterCard SecureCode). MasterCard '**1**' - Cardholder not participating. MasterCard '**2**' - Cardholder authenticated. Visa '**05**' - Fully Authenticated. Visa '**06**' - Not authenticated (cardholder not participating). Visa '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid. American Express '**05**' - Fully Authenticated. American Express '**06**' - Not authenticated (cardholder not participating). American Express '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid.			
Output	Numeric	0,2	06

Mode 3b - 2-Party Style Pre-Authenticated Payment

The following additional inputs are added to a standard 2-Party Authorisation or Purchase transaction where the cardholder has already been pre-Authenticated in a Mode 3a operation.

Mode 3b Transaction Request Input Fields

Pre Authentication Payment Fields			
The data is sent by simply including the additional data with the required fields for a basic transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_VerType			
This field must be a value of '**3DS**' for the following fields to operate			
Required	Alphanumeric	3	3DS

vpc_VerToken			
This value is generated by the Access Control Server at the card issuer as a token to prove that the cardholder authenticated OK. This is a base64 encoded value.			
Required	Alphanumeric	28	gIGCg4SFhoeIiYqLjI2Oj5CRkpM=

vpc_3DSXID			
It is a unique transaction identifier that is generated by the Payment Server on behalf of the merchant to identify the 3DS transaction. It is a 20-byte field that is Base64 encoded to produce a 28-character value.			
Required	Alphanumeric	28	HA1r1v2kDghhQw9DMQi/wQacCL8=

vpc_3DSECI			
It is the 3-D Secure Electronic Commerce Indicator, which is returned from the Issuers ACS. For Verified by Visa and American Express SafeKey, this is '05' where the Issuers ACS has validated the cardholders password or '06' where an 'Attempts ACS' condition has occurred. For Mastercard SecureCode, if OK the value will be either '01' or '02', and '06' when the cardholder attempts to authenticate. (These values may change depending on the locale or issuer).			
Required	Alphanumeric	2	05

vpc_3DSenrolled			
This field is mandatory if the card is within an enrolled range. This is the value of the VERes.enrolled field. It will take values (**Y** - Yes, **N** - No, **U** - Unavailable for Checking).			
Conditional	Alphanumeric	1	Y

vpc_3DSstatus			
This field is only included if 3-D Secure authentication was used and a PARes was received by the MPI. It will take values (**Y** - Yes, **A** - Attempted Authentication, **U** - Unavailable for Checking).			

Conditional	Alphanumeric	1	Y

Mode 3b Transaction Response Output Fields

Payment Authentication Output Fields			
In addition to the standard output fields, the following fields are also returned in the Transaction Response for 2-Party pre-Authenticated transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_3DSECI			
The 3-D Secure Electronic Commerce Indicator, which is set to '05' when the cardholder authenticates OK, and '07' when the cardholder is not enrolled. (These values may change depending on the locale or issuer).			
Output	Numeric	0,2	07

vpc_3DSXID			
It is a unique transaction identifier that is generated by the Payment Server on behalf of the merchant to identify the 3DS transaction. It is a 20-byte field that is Base64 encoded to produce a 28-character value.			
Output	Alphanumeric	0,28	uyPfGIgsoFQhklkIsto+IFWs92s=

vpc_3DSenrolled			
This field indicates if the card is within an enrolled range. This is the value of the VERes.enrolled field. It will take values (**Y** - Yes, **N** - No, **U** - Unavailable for Checking).			
Output	Alpha	1	N

vpc_3DSstatus			
This field is only included if payment authentication was attempted and a PARes was received by the MPI. It will take values (**Y** - Yes, **N** - No, **A** - Attempted Authentication, **U** - Unavailable for Checking).			
Output	Alpha	0,1	N

vpc_VerToken			
This value is generated by the card issuer as a token to prove that the cardholder authenticated OK. This is a base64 encoded value.			
Output	Alphanumeric	28	gIGCg4SFhoeIiYqLjI2Oj5CRkpM=

vpc_VerType			
This field will either be '**3DS**' 3-D Secure incorporating one of the 3-D Secure schemes. For example, Verified by Visa or MasterCard SecureCode or '**SPA**' - Secure Payment Authentication from MasterCard (rarely used).			
Output	Alphanumeric	0,3	3DS

vpc_VerStatus

The status codes used by the Payment Server to show whether the payment authentication was successful or not. ***3-D Secure Status Codes*** on page 128.			
Output	Alphanumeric	1	N

vpc_VerSecurityLevel			
The Verification Security Level is generated at the card issuer as a token to prove that the cardholder was enrolled and authenticated OK. It is shown for all transactions except those with authentication status "Failure". This field contains the security level to be used in the AUTH message. MasterCard '**0**' - Merchant not participating (a merchant will not see this if they are configured for MasterCard SecureCode). MasterCard '**1**' - Cardholder not participating. MasterCard '**2**' - Cardholder authenticated. Visa '**05**' - Fully Authenticated. Visa '**06**' - Not authenticated (cardholder not participating). Visa '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid. American Express '**05**' - Fully Authenticated. American Express '**06**' - Not authenticated (cardholder not participating). American Express '**07**' - Not authenticated. Usually due to a system problem, for example the merchant password is invalid.			
Output	Numeric	0,2	06

CHAPTER 5

Advanced Merchant Administration (AMA) Transactions

Advanced Merchant Administration (AMA) is used when the volume of transactions is too great to be economically viable or too difficult to be carried out manually. AMA transactions allow the merchant to incorporate additional features such as refunds, into the merchant system. All of these transactions operate using the 2-Party model.

Capture, Refund, Void Capture, Void Refund and Void Purchase return standard output fields, plus a comma (',') delimited result string containing a host of other data.

Note: Some financial institutions do not support voids.

Merchants and users who need AMA transactions must have a username and password; in addition, they must be set up with the appropriate AMA privileges to run a particular AMA transaction.

Note: Applies to 2-Party transactions.

An AMA user cannot be used for Merchant Administration operations.

CHAPTER 6

Basic Transaction Fields

This section describes the commands, field types and valid values for basic transactions in Virtual Payment Client.

Basic Input Fields - AMA Transaction

Data is sent from the merchant application to the Payment Server via the Virtual Payment Client, a basic transaction requiring a number of data fields as per the table below.

The fields are sent to a fully qualified URL (starting with HTTPS://) via a HTTP POST operation. This URL must be included in the merchant's application code to send transaction information to the Virtual Payment Client.

https://<YOUR_VPC_URL>/vpcdps

Note: This URL is supplied by the Payment Provider.

2-Party AMA Input Fields			
The following data fields must be included in a Transaction Request when using a 2-Party AMA transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Version			
The version of the Virtual Payment Client API being used. The current version is 1.			
Required	Alphanumeric	1,8	1
vpc_AccessCode			
Authenticates the merchant on the Payment Server. This means that a merchant cannot access another merchant's Merchant Id. The access code is provided when the merchant profile is registered with a Payment Provider.			
Required	Alphanumeric	8	6AQ89F3
vpc_MerchTxnRef			
A unique value created by the merchant. **Usage Notes:** The Merchant Transaction Reference is used as a reference key to the Payment Server database to obtain a copy of lost/missing transaction receipts using the QueryDR function. It can also be used to identify a duplicate transaction if it is always kept unique for each transaction attempt. It can contain similar information to the vpc_OrderInfo field, but it must be unique for each transaction attempt if it is to be used properly. Typically, the vpc_MerchTxnRef is based on an order number, invoice number, timestamp, etc., but it should also reflect the transaction attempt. For example, if a cardholder has insufficient funds on their card and they are allowed to repeat the transaction with another credit card, the value may be INV1234/1 on the first attempt, INV1234/2 on the second attempt, and INV1234/3 on the third attempt. This identifier will be displayed in the Transaction Search results and also in the Download file (from Financial Transactions Search or Download Search Results link in Financial Transaction List) in the Merchant Administration portal on the Payment Server. **Note**: If "Enforce Unique Merchant Transaction Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's transactions.			
Required	Alphanumeric	1,40	ORDER958743-1

vpc_Merchant			
The unique Merchant Id assigned to a merchant by the Payment Provider. The Merchant ID identifies the merchant account against which settlements will be made.			
Required	Alphanumeric	1,16	TESTMERCHANT01
vpc_TransNo			
This is the unique Payment Server Order Number (Shopping Transaction Number) generated by the Payment Server for the initial transaction.			
Required	Numeric	1,19	10712
vpc_User			
The user name of the user who is performing the AMA transaction. Each AMA User name may be assigned different privileges to perform particular functions. For example, an AMA User can be set to only perform refunds. **Note:** An AMA user cannot be used for Merchant Administration operations.			
Required	Alphanumeric	1,20	Maryellen
vpc_Password			
The password used by the merchant to authorise Advanced Merchant Administration transactions. It must be at least 8 characters long and contain at least one non-alphabetical character.			
Required	Alphanumeric	8,25	T1m34t*A

Basic Output Fields - AMA Transaction

Once a Transaction Response has been successfully received, the merchant application can retrieve the receipt details. These values are then passed back to the cardholder for their records.

Note: The Transaction Response provided by the Payment Server may contain other fields that are not documented in this guide. Such fields may be changed, added, or removed without notice, and must NOT be relied upon by merchant integrations.

Terminology: Returned Input fields are shown as "Input" in the table.

2-Party AMA Output Fields			
The following data fields are returned in a Transaction Response for a standard 2-Party transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Version			
The version of the Virtual Payment Client API being used. The current version is 1.			
Input	Alphanumeric	1,8	1
vpc_Command			
The value of the vpc_Command input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	pay
vpc_Locale			
Specifies the language used on the Payment Server based on your merchant configuration.			
Input	Alpha	2,5	en
vpc_MerchTxnRef			
The value of the vpc_MerchTxnRef input field returned in the Transaction Response. This field may not be returned in a transaction that fails due to an error condition.			
Input	Alphanumeric	0,40	ORDER958743-1
vpc_Merchant			
The value of the vpc_Merchant input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	TESTMERCHANT01
vpc_Message			
This is a message to indicate what sort of errors the transaction encountered. This field is not provided if vpc_TxnResponseCode has a value of zero.			
Output	Alphanumeric	1,255	Merchant [TESTCORE23] does not exist.
vpc_TxnResponseCode			
A response code that is generated by the Payment Server to indicate the status of the transaction. A vpc_TxnResponseCode of “0” (zero) indicates that the transaction was processed successfully and approved by the acquiring bank. Any other value indicates that the transaction was declined (it went through to the banking network) or the transaction failed (it never made it to the banking network). For a list of values, see Transaction Response Codes.			

Output	Alphanumeric	1	0
vpc_AcqResponseCode			
Generated by the financial institution to indicate the status of the transaction. The results can vary between institutions so it is advisable to use the vpc_TxnResponseCode as it is consistent across all acquirers. It is only included for fault finding purposes. Most Payment Providers return the vpc_AcqResponseCode as a 2-digit response, others return it as a 3-digit response. This field is not returned for transactions that result in an error condition.			
Output	Alphanumeric	2,3	00
vpc_TransactionNo			
Financial Transaction Number is a unique number generated by the Payment Server for this transaction. This field will not be returned if the transaction failed due to an error condition.			
Output	Numeric	1,19	96841
vpc_BatchNo			
A value supplied by an acquirer which indicates the batch of transactions that the specific transaction has been grouped with. Batches of transactions are settled by the acquirer at intervals determined by them. This is an acquirer specific field, for example, it could be a date in the format YYYYMMDD. This field will not be returned if the transaction fails due to an error condition.			
Output	Numeric	0,8	20060105
vpc_AuthorizeId			
Authorisation Identification Code issued by the Acquirer to indicate the approval of a transaction. This field is 6-digits maximum and is not returned for transactions that are declined or fail due to an error condition. **Note**: This field may not be returned based on the transaction type and your acquirer configuration.			
Output	Alphanumeric	0,6	654321
vpc_ReceiptNo			
A unique identifier that is also known as the Reference Retrieval Number (RRN). The vpc_ReceiptNo may be passed back to the cardholder for their records if the merchant application does not generate its own receipt number. This field is not returned for transactions that result in an error condition.			
Output	Alphanumeric	0,12	RP12345
vpc_Amount			
The value of the vpc_Amount input field returned in the Transaction Response. For Void transactions, vpc_Amount indicates the amount associated with the Order you wish to void.			
Input	Numeric	1,10	1250
vpc_Card			
Identifies the card type used for the transaction. For a list of card types see Card Type Codes. This field is not returned for transactions that result in an error condition.			
Output	Alpha	0,2	MC
vpc_Currency			
The value of the vpc_Currency input field returned in the Transaction Response. This field is returned only if vpc_Currency was included in the Transaction Request.			
Input	Alpha	3	USD

vpc_ShopTransactionNo			
This is the unique Payment Server Order Number (Shopping Transaction Number) generated by the Payment Server for the initial transaction.			
Input	Numeric	1,19	10712
vpc_TicketNumber			
The ticket number was originally aimed at the airline industry, however it can be used for any relevant information about this transaction you want stored. The ticket number is stored on the Payment Server database for that transaction and returned in the Transaction Response for capture transactions. This field is only returned if <Input_TicketNumber> was supplied in the initial transaction.			
Output	Alphanumeric	0,15	VIP Client
vpc_AcqResponseText			
The response from the acquirer in the text form. This field is used instead of vpc_AcqResponseCode for acquirers that return text instead of a single code.			
Optional	Alphanumeric	0,255	Success : Pending: Authorization
vpc_TerminalID			
Specifies the terminal ID used to process the transaction with your acquirer.			
Output	Alphanumeric	4,8	123456

AMA Capture Transaction

The AMA Capture command allows a merchant to capture the funds from a previous authorisation transaction.

Transaction Request Input Fields

2-Party Capture Input Fields			
The following data fields must be included in a Transaction Request when performing a Capture transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the command type. This must be equal to '**capture**' for a capture transaction.			
Required	Alphanumeric	1,16	capture

vpc_Amount			
The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.			
Required	Numeric	1,12	1250
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD

Transaction Response Output Fields

2-Party Capture Output Fields
The following additional data fields are returned in a Transaction Response for a Capture transaction.
Field Name
Field Description

Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_AuthorisedAmount			
This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10185
vpc_CapturedAmount			
This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10100
vpc_RefundedAmount			
This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	1,10	1295

AMA Refund Transaction

AMA Refund allows you to refund funds for a previous purchase or capture transaction from the merchant's account back to the cardholder's account.

Transaction Request Input Fields

2-Party Refund Input Fields			
The following fields must be included in a Transaction Request when performing a Refund transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**refund**' for a refund transaction.			
Required	Alphanumeric	1,16	refund

vpc_Amount			
The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.			
Required	Numeric	1,12	1250
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD
vpc_FinTransNo			
The financial transaction number of the capture you wish to refund. This field only needs to be supplied when more than one capture exists for an order.			
Note: This field is currently applicable only to PayPal transactions. For all other non-PayPal transactions, this field will be ignored.			
Optional	Numeric	1,19	10712

Transaction Response Output Fields

2-Party Refund Output Fields			
The following additional data fields are returned in a Transaction Response for a Refund transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_AuthorisedAmount			
This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10185
vpc_CapturedAmount			
This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10100
vpc_RefundedAmount			
This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	1,10	1295

AMA Void AuthorisationTransaction

AMA Void Authorisation allows a merchant to void the authorisation from a previous authorisation transaction in Auth/Capture mode, that has not been processed by the acquiring institution.

Transaction Request Input Fields

2-Party Void Authorisation Input Fields			
The following data fields must be included in a Transaction Request when using for a Void Capture transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**voidAuthorisation**' for a void authorisation transaction.			
Required	Alphanumeric	1,16	voidAuthorisation
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

AMA Void Capture Transaction

AMA Void Capture allows a merchant to void the funds from a previous capture transaction in Auth/Capture mode, that has not been processed by the acquiring institution.

Transaction Request Input Fields

2-Party Void Capture Input Fields			
The following data fields must be included in a Transaction Request when using for a Void Capture transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**voidCapture**' for a void capture transaction.			
Required	Alphanumeric	1,16	voidCapture
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD

Transaction Response Output Fields

2-PartyVoid Capture Output Fields			
The following additional data fields are returned in a Transaction Response for a Void Capture ransaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_AuthorisedAmount			
This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10185
vpc_CapturedAmount			
This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10100

vpc_RefundedAmount			
This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	1,10	1295

AMA Void Purchase Transaction

AMA Void Purchase allows a purchase merchant to void a purchase transaction that has not been processed by the acquiring institution. It is not available for Auth/Capture mode merchants.

Transaction Request Input Fields

2-Party Void Purchase Input Fields			
The following data fields must be included in a Transaction Request when using for a Void Purchase transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**voidPurchase**' for this transaction type.			
Required	Alphanumeric	1,16	voidPurchase
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD

Transaction Response Output Fields

2-PartyVoid Purchase Output Fields			
The following additional data fields are returned in a Transaction Response for a Void Purchase transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_AuthorisedAmount			
This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10185
vpc_CapturedAmount			
This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10100

vpc_RefundedAmount			
This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	1,10	1295

AMA Void Refund Transaction

AMA Void Refund allows a merchant to void a previous refund transaction that has not been processed by the acquiring institution.

Transaction Request Input Fields

2-Party Void Refund Input Fields			
The following data fields must be included in a Transaction Request when using for a Void Refund transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Command			
Indicates the transaction type. This must be equal to '**voidRefund**' for this transaction type.			
Required	Alphanumeric	1,16	voidRund
vpc_Currency			
The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only. This value must match the currency of the existing order that is being identified by vpc_TransNo.			
Optional	Alpha	3	USD

Transaction Response Output Fields

2-PartyVoid Refund Output Fields			
The following additional data fields are returned in a Transaction Response for a Void Refund transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_AuthorisedAmount			
This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10185
vpc_CapturedAmount			
This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	0,10	10100
vpc_RefundedAmount			

This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.			
Output	Numeric	1,10	1295

AMA Standalone Capture Transaction

Standalone Capture allows you to capture funds against an order when the corresponding authorisation was obtained either manually, or in an external system.

Use the Standalone Capture command via the Virtual Payment Client to directly perform captures from your application. Your Payment Provider must enable this function on your Merchant Profile for you to use this functionality.

Transaction Request Input Fields

2-Party Standalone Capture Input Fields			
The following data fields must be included in a Transaction Request when performing a Standalone Capture transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**doRequest**' for this type of transaction.			
Required	Alphanumeric	1,16	doRequest

vpc_RequestType			
This field is associated when the **vpc_Command** field equals '**doRequest**'. The value must be equal to '**CAPTURE**' for this type of transaction.			
Required	Alphanumeric	1,20	CAPTURE

vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. Applicable values can be obtained from your Payment Services Provider. The value must be equal to '**doStandaloneCapture**' for this type of transaction.			
Required	Alphanumeric	1,20	doStandaloneCapture

vpc_OrderInfo			
The merchant's identifier used to identify the order on the Payment Server. For example, a shopping cart number, an order number, or an invoice number. This identifier will be displayed in the Transaction Search results in the Merchant Administration portal on the Payment Server.			
Note: If 'Enforce Unique Order Reference'' privilege is enabled by your Payment Provider, this value must be unique across all the merchant's orders.			
Required	Alphanumeric	0,34	ORDER958743

vpc_ManualAuthID

<table>
<tr><td colspan="4">An alphanumeric code of up to six characters used to specify the manual authorisation code supplied by the card issuer for the transaction.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>0,6</td><td>AB3456</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_CardNum</td></tr>
<tr><td colspan="4">The number of the card used for the transaction. The format of the Card Number is based on the Electronic Commerce Modeling Language (ECML) and, in particular, must not contain white space or formatting characters.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>15,19</td><td>5123456789012346</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_CardExp</td></tr>
<tr><td colspan="4">The expiry date of the card in the format YYMM. The value must be expressed as a 4-digit number (integer) with no white space or formatting characters. For example, an expiry date of May 2013 is represented as 1305.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>4</td><td>1305</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_CardIssueNumber</td></tr>
<tr><td colspan="4">The issue number of the card used with cards such as Maestro and Solo.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo.</td></tr>
<tr><td>Optional</td><td>Numeric</td><td>0,2</td><td>01</td></tr>
<tr><td colspan="4">vpc_CardStartDate</td></tr>
<tr><td colspan="4">The start date of the card in yymm format used with cards such as Maestro and Solo. The value must be expressed as a 4-digit number (integer) with no white spaces or formatting characters. For example, an expiry date of May 2013 is represented as 1305.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo.</td></tr>
<tr><td>Optional</td><td>Numeric</td><td>4</td><td>1305</td></tr>
<tr><td colspan="4">vpc_BankAccountType</td></tr>
<tr><td colspan="4">The type of bank account the cardholder wants to use for the transaction. For example, Savings or Cheque.
Valid values for this field are:
CHQ — specifies that the cardholder wants to use the Cheque account linked to the card.
SAV — specifies that the cardholder wants to use the Savings account linked to the card.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo, and will be displayed in the Order Search results in the Merchant Administration portal on the Payment Server.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>3</td><td>SAV</td></tr>
<tr><td colspan="4">vpc_Currency</td></tr>
<tr><td colspan="4">The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only.
The merchant must be configured to accept the currency used in this field. To obtain a list of supported currencies and codes, please contact your Payment Provider.</td></tr>
<tr><td colspan="4"><b>Note</b>: This field is required only if more than one currency is configured for the merchant.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>3</td><td>USD</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_Amount</td></tr>
</table>

<table>
<tr><td colspan="4">The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250.
This value cannot be negative or zero. The maximum valid value is 2147483647.</td></tr>
<tr><td colspan="4"><b>Note</b>: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>1,12</td><td>1250</td></tr>
</table>

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

AMA Standalone Refund Transaction

Standalone Refund allows you to refund funds from your account back to the cardholder without a previous purchase.

Use the Standalone Refund command via the Virtual Payment Client to directly perform refunds from your application. Your Payment Provider must enable this function on your Merchant Profile for you to use this functionality.

Transaction Request Input Fields

2-Party Standalone Refund Input Fields			
The following data fields must be included in a Transaction Request when performing transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**doRequest**' for this type of transaction.			
Required	Alphanumeric	1,16	doRequest

vpc_RequestType			
This field is associated when the **vpc_Command** field equals '**doRequest**'. The value must be equal to '**CREDIT**' for this type of transaction.			
Required	Alphanumeric	1,20	CREDIT

vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. Applicable values can be obtained from your Payment Services Provider. The value must be equal to '**doStandaloneRefund**' for this type of transaction.			
Required	Alphanumeric	1,20	doStandaloneRefund

vpc_OrderInfo			
The merchant's identifier used to identify the order on the Payment Server. For example, a shopping cart number, an order number, or an invoice number. This identifier will be displayed in the Transaction Search results in the Merchant Administration portal on the Payment Server.			
Note: If 'Enforce Unique Order Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's orders.			
Required	Alphanumeric	0,34	ORDER958743
vpc_CardNum			

<table>
<tr><td colspan="4">The number of the card used for the transaction. The format of the Card Number is based on the Electronic Commerce Modeling Language (ECML) and, in particular, must not contain white space or formatting characters.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>15,19</td><td>5123456789012346</td></tr>
<tr><td colspan="4">vpc_CardExp</td></tr>
<tr><td colspan="4">The expiry date of the card in the format YYMM. The value must be expressed as a 4-digit number (integer) with no white space or formatting characters. For example, an expiry date of May 2013 is represented as 1305.</td></tr>
<tr><td>Required</td><td>Numeric</td><td>4</td><td>1305</td></tr>
<tr><td colspan="4">vpc_CardSecurityCode</td></tr>
<tr><td colspan="4">The Card Security Code (CSC), also known as CVV(Visa), CVC2(MasterCard) or CID/4DBC(Amex) or CVV2, which is printed, not embossed on the card. It compares the code with the records held in the card issuing institution's database.</td></tr>
<tr><td>Optional</td><td>Numeric</td><td>3,4</td><td>985</td></tr>
<tr><td colspan="4">vpc_CardStartDate</td></tr>
<tr><td colspan="4">The start date of the card in yymm format used with cards such as Maestro and Solo. The value must be expressed as a 4-digit number (integer) with no white spaces or formatting characters. For example, an expiry date of May 2013 is represented as 1305.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo.</td></tr>
<tr><td>Optional</td><td>Numeric</td><td>4</td><td>1305</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_CardIssueNumber</td></tr>
<tr><td colspan="4">The issue number of the card used with cards such as Maestro and Solo.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo.</td></tr>
<tr><td>Optional</td><td>Numeric</td><td>0,2</td><td>01</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_BankAccountType</td></tr>
<tr><td colspan="4">The type of bank account the cardholder wants to use for the transaction. For example, Savings or Cheque.
Valid values for this field are:
CHQ — specifies that the cardholder wants to use the Cheque account linked to the card.
SAV — specifies that the cardholder wants to use the Savings account linked to the card.</td></tr>
<tr><td colspan="4"><b>Usage Notes:</b> This identifier is mandatory if the card type is Maestro or Solo, and will be displayed in the Order Search results in the Merchant Administration portal on the Payment Server.</td></tr>
<tr><td>Optional</td><td>Alphanumeric</td><td>3</td><td>SAV</td></tr>
<tr><td colspan="4">vpc_Currency</td></tr>
<tr><td colspan="4">The currency of the order expressed as an ISO 4217 alpha code. This field is case-sensitive and must include uppercase characters only.
The merchant must be configured to accept the currency used in this field. To obtain a list of supported currencies and codes, please contact your Payment Provider.</td></tr>
<tr><td colspan="4"><b>Note</b>: This field is required only if more than one currency is configured for the merchant.</td></tr>
<tr><td>Optional</td><td>Alpha</td><td>3</td><td>USD</td></tr>
</table>

<table>
<tr><td>vpc_Amount</td></tr>
</table>

The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.			
Required	Numeric	1,12	1250

Transaction Response Output Fields

There are no special output fields returned in the Transaction Response.

AMA AVS Only Transaction

AMA Address Verification Service (AVS) Only is a security feature, which allows a merchant to verify the cardholder's address details.

With AVS Only, the Payment Server strips the value in the Authorisation transaction and substitutes a nominal transaction value (usually $1.00). The acquiring bank checks the card details with the issuing card institution to ensure they are correct. No funds at all are reserved on the card.

The issuer returns an AVS result code to indicate the level of match of the address provided by the merchant. It is then up to the merchant to determine whether to proceed with the transaction.

Transaction Request Input Fields

2-Party AVS Only Input Fields			
The following data fields must be included in a Transaction Request when using for a Void Capture transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Command			
Indicates the transaction type. This must be equal to '**doRequest**' for this type of transaction.			
Required	Alphanumeric	1,16	doRequest

vpc_RequestType			
This field is associated when the **vpc_Command** field equals '**doRequest**'. This must be equal to '**AVS**' for this type of transaction.			
Required	Alphanumeric	1,20	AVS
vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. Applicable values can be obtained from your Payment Services Provider. The value must be equal to '**doAvsOnly**' for this type of transaction.			
Required	Alphanumeric	1,20	doAvsOnly

vpc_CardSecurityCode			
The Card Security Code (CSC), also known as CVV(Visa), CVC2(MasterCard) or CID/4DBC(Amex) or CVV2, which is printed, not embossed on the card. It compares the code with the records held in the card issuing institution's database.			
Optional	Numeric	3,4	985

vpc_OrderInfo

The merchant's identifier used to identify the order on the Payment Server. For example, a shopping cart number, an order number, or an invoice number. This identifier will be displayed in the Transaction Search results in the Merchant Administration portal on the Payment Server.			
Note: If 'Enforce Unique Order Reference" privilege is enabled by your Payment Provider, this value must be unique across all the merchant's orders.			
Required	Alphanumeric	0,34	ORDER958743

vpc_Amount			
The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, ∃12.50 is expressed as 1250. This value cannot be negative or zero. The maximum valid value is 2147483647.			
Note: Transactions in currency IDR (Indonesian Rupiah) will use an exponent of 0 (zero). This means an amount expressed as 1250 will be treated as IDR Rp1,250 and not IDR Rp12.50 (with exponent 2) unlike other currencies.			
Required	Numeric	1,12	1250

vpc_AVS_Street01			
The street name and number, or the Post Office Box details, of the address used in the credit card billing Address Verification check by the card issuing bank.			
Required	Alphanumeric	1,128	1136 John Street
vpc_AVS_City			
The city/town/village of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alphanumeric	1,128	Seattle
vpc_AVS_StateProv			
The State/Province code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alphanumeric	0,128	WA
vpc_AVS_PostCode			
The Postal/Zip code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Required	Alphanumeric	4,9	98111
vpc_AVS_Country			
The 3 digit ISO standard alpha country code of the address used in the credit card billing Address Verification check by the card issuing bank.			
Optional	Alpha	3	USA

Transaction Response Output Fields

Once a Transaction Response has been successfully received, the merchant application can retrieve the receipt details. These values are then passed back to the cardholder for their records.

2-Party AVS Only Output Fields

<table>
<tr><td colspan="4">The following data fields are returned in a Transaction Response for standard transactions.</td></tr>
<tr><td colspan="4">Field Name</td></tr>
<tr><td colspan="4">Field Description</td></tr>
<tr><td>Returned Input or Output</td><td>Field Type</td><td>Min, Max or Set Field Length</td><td>Sample Data</td></tr>
<tr><td colspan="4">vpc_AVS_Street01</td></tr>
<tr><td colspan="4">The value of the vpc_AVS_Street01 input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>0,20</td><td>1136 John Street</td></tr>
<tr><td colspan="4">vpc_AVS_City</td></tr>
<tr><td colspan="4">The value of the vpc_AVS_City input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>0,20</td><td>Seattle</td></tr>
<tr><td colspan="4">vpc_AVS_StateProv</td></tr>
<tr><td colspan="4">The value of the vpc_AVS_StateProv input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>0,5</td><td>WA</td></tr>
<tr><td colspan="4">vpc_AVS_PostCode</td></tr>
<tr><td colspan="4">The value of the vpc_AVS_PostCode input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>0,9</td><td>98111</td></tr>
<tr><td colspan="4">vpc_AVS_Country</td></tr>
<tr><td colspan="4">The value of the vpc_AVS_Country input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alpha</td><td>0,3</td><td>USA</td></tr>
<tr><td colspan="4">vpc_AVSResultCode</td></tr>
<tr><td colspan="4">The result code generated by the Payment Sever to indicate the AVS level that was used to match the data held by the cardholder's issuing bank. For more information, see <b><i>AVS Result Codes</i></b> on page 121.
<b>Note:</b> It can also be returned as '<b>Unsupported</b>' if the acquirer does not support this field.</td></tr>
<tr><td>Output</td><td>Alpha</td><td>1,11</td><td>Y</td></tr>
<tr><td colspan="4">vpc_AcqAVSRespCode</td></tr>
<tr><td colspan="4">Generated by the card issuing institution in relation to AVS. Provided for ancillary information only.</td></tr>
<tr><td>Output</td><td>Alpha</td><td>1,11</td><td>Y</td></tr>
</table>

AMA QueryDR

The AMA QueryDR command allows a merchant to search for the current or the most recent transaction receipt. It also queries for unknown transactions (a transaction request that was never received) and failed transactions. The search is performed on the key - *vpc_MerchTxnRef*, so the *vpc_MerchTxnRef* field must be a unique value. If more than one Transaction Response exists with the same *vpc_MerchTxnRef,* the most recent Transaction Response is returned. For QueryDR to return the current transaction, the transaction response code of the original Transaction Response must be "P-Pending" or "M-Submitted".

If you want to use QueryDR to return digital receipts, it must be done in under 3 days or no results matching the criteria will be returned. This is because the database only contains data up to 3 days old.

Transaction Request Input Fields

2-Party QueryDR Input Fields			
The following data fields must be included in a Transaction Request when using a QueryDR check.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_Command			
Indicates the transaction type. This must be equal to '**queryDR**' for a QueryDR function.			
Required	Alphanumeric	1,16	queryDR

Transaction Response Output Fields

A QueryDR can be performed on on a base transaction, or on AMA transactions such as a Capture, Refund or Void. Both of these transaction types return different fields.

QueryDR Output Fields			
The following additional data fields are returned in a Transaction Response for a QueryDR transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_DRExists			
This key is used to determine if the QueryDR command returned any search results. If the value is "**Y**", there is one transaction with a MerchTxnRef number that matched the search criteria. If the value is "**N**", then there is no matching MerchTxnRef number result for the search criteria.			
Output	Alpha	1	Y

vpc_FoundMultipleDRs			
This is used after the previous command to determine if there are multiple results. If the value is "**Y**", there are multiple transactions with the MerchTxnRef number that matches the search criteria. If the value is "**N**", there could be zero or at most, one transaction with the MerchTxnRef number that matches the search criteria.			
Output	Alpha	1	N

If an original receipt exists, the QueryDR will return all the ***basic AMA output fields*** on page 88 in addition to vpc_DRExists and vpc_FoundMultipleDRs. If the transaction to be queried is a subsequent/AMA transaction such as Capture, Refund, or Void then the following additional fields are returned.

<table>
<tr><td colspan="4">vpc_AuthorisedAmount</td></tr>
<tr><td colspan="4">This is the value of the Authorised transaction amount for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.</td></tr>
<tr><td>Output</td><td>Numeric</td><td>0,10</td><td>10185</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_CapturedAmount</td></tr>
<tr><td colspan="4">This is the value of the total transaction amount captured for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.</td></tr>
<tr><td>Output</td><td>Numeric</td><td>0,10</td><td>10100</td></tr>
<tr><td colspan="4">vpc_RefundedAmount</td></tr>
<tr><td colspan="4">This is the total value of any Refunded transaction amounts for the order and is returned in the Transaction Response of a Capture, Refund and Void transaction amount for Virtual Payment Client.</td></tr>
<tr><td>Output</td><td>Numeric</td><td>1,10</td><td>1295</td></tr>
</table>

If an original receipt doesn't exist, the QueryDR will return the following fields in addition to vpc_DRExists and vpc_FoundMultipleDRs.

<table>
<tr><td colspan="4">vpc_Version</td></tr>
<tr><td colspan="4">The version of the Virtual Payment Client API being used. The current version is 1.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>1,8</td><td>1</td></tr>
</table>

<table>
<tr><td colspan="4">vpc_Amount</td></tr>
<tr><td colspan="4">The value of the vpc_Amount input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Numeric</td><td>1,10</td><td>1250</td></tr>
<tr><td colspan="4">vpc_BatchNo</td></tr>
<tr><td colspan="4">A value supplied by an acquirer which indicates the batch of transactions that the specific transaction has been grouped with. Batches of transactions are settled by the acquirer at intervals determined by them.
This is an acquirer specific field, for example, it could be a date in the format YYYYMMDD.
This field will not be returned if the transaction fails due to an error condition.</td></tr>
<tr><td>Output</td><td>Numeric</td><td>0,8</td><td>20060105</td></tr>
<tr><td colspan="4">vpc_Command</td></tr>
<tr><td colspan="4">The value of the vpc_Command input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alphanumeric</td><td>1,16</td><td>pay</td></tr>
<tr><td colspan="4">vpc_Locale</td></tr>
<tr><td colspan="4">The value of the vpc_Locale input field returned in the Transaction Response.</td></tr>
<tr><td>Input</td><td>Alpha</td><td>2,5</td><td>en</td></tr>
<tr><td colspan="4">vpc_Merchant</td></tr>
</table>

The value of the vpc_Merchant input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	TESTMERCHANT01
vpc_TransactionNo			
Financial Transaction Number is a unique number generated by the Payment Server for this transaction. This field will not be returned if the transaction failed due to an error condition.			
Output	Numeric	1,19	96841

CHAPTER 7

References - Virtual Payment Client

Generating a Secure Hash

Note: New merchant integrations are required to generate a secure hash using the SHA-256 HMAC algorithm.

Creating a SHA-256 HMAC Secure Hash

The merchant code creates the Secure Hash value on the Transaction Request data. The Payment Server creates another Secure Hash value and sends it back to the merchant in the Transaction Response.

The Secure Hash is a hexadecimal encoded SHA-256 HMAC of a concatenation of VPC and User Defined parameters. The concatenation of parameters takes the form of a set of name-value pairs, similar to the parameter string for an HTTP GET call.

Merchant- Supplied Parameters

For information that you want to return to your integration in the Transaction Response, you may either:

- Include it in an appropriate VPC parameter such as vpc_MerchTxnRef field or vpc_ReturnURL in the Transaction Request, or
- Provide User Defined parameters in the Transaction Request. User Defined parameters are identified by having a parameter name starting with "user_". These fields should be used in the SHA-256 HMAC calculation.
- Provide other Merchant Supplied parameters. Other Merchant Supplied parameters (that do not begin with “user_”) are not included in the SHA-256 HMAC calculation.

Note: All field names are restricted to the character set defined by the regular expression [A-Za-z0-9_].

SHA-256 HMAC Calculation

The SHA-256 HMAC is calculated as follows:

1 The SHA-256 HMAC calculation includes all VPC and User Defined fields, that is all fields beginning with "vpc_" and "user_", except the vpc_SecureHash and vpc_SecureHashType parameters.

   The field names are sorted in ascending order of parameter name. Specifically, the sort order is:

   - ascending order of parameter name using the ASCII collating sequence, for example, "Card" comes before "card"
   - where one string is an exact substring of another, the smaller string should be ordered before the longer, for example, "Card" should come before "CardNum".

2 Construct a string by concatenating the string form of the sorted field name-value pairs. The string form of a name-value pair is the name followed by the value.

   - The field name and the value in each field name-value pair are joined using "=" as the separator.
   - The resulting joined field name-value pairs are themselves joined using "&" as the separator.

3 Create a SHA-256 HMAC of the resultant string using the **hex decoded** value of your merchant secret as the key. The SHA-256 HMAC algorithm is defined in Federal Information Processing Standard 180-2. We strongly recommend that you use one of the numerous implementations available in most programming languages.

> **Note**: It is **critical** that you use the hex decoded value of the secret as the key. For example, in PHP you can use the `pack('H*',SecureSecret)` function. In C# you will need to create and parse a byte array as demonstrated in the example code.

4 Encode the HMAC in hexadecimal, and include it in the request as the value for the vpc_SecureHash field.

For example, if your merchant secret is BB48A64077A1CBF08FF0D91C5A9FE42B, and the Transaction Request includes only the following parameters:

Field Name	Example Value
vpc_Version	1
vpc_Command	pay
vpc_MerchTxnRef	txn1
vpc_CardNum	345678901234564
vpc_CardExp	1305
vpc_Merchant	TNSITESTMERCHANT
vpc_AccessCode	75A6GH9
vpc_Amount	1000
user_SessionId	567890

The concatenated value is as follows:

```
user_SessionId=567890&vpc_AccessCode=75A6GH9&vpc_Amount=1000&vpc_CardExp=1305&vpc_C
ardNum=345678901234564&vpc_Command=pay&vpc_MerchTxnRef=txn1&vpc_Merchant=TNSITESTME
RCHANT&vpc_Version=1
```

> **Note 1**: The last character of each field value (other than the last) is followed directly by "&". The concatenated value must be represented in the UTF-8 character encoding format.
>
> **Note 2**: The values in all name value pairs should NOT be URL encoded for the purpose of hashing.

The Secure Hash value is:

```
3812B7C7D21726AAC9633E1D42BD43A73A329F8906C248EFAF9CEC354F8B0C08
```

And the resultant Request is (note the Secure Hash and Secure Hash Type fields):

```
user_SessionId=567890&vpc_AccessCode=75A6GH9&vpc_Amount=1000&vpc_CardExp=1305&vpc_C
ardNum=345678901234564&vpc_Command=pay&vpc_MerchTxnRef=txn1&vpc_Merchant=TNSITESTME
RCHANT&vpc_Version=1&vpc_SecureHash=7C6866D0B1DF14FE03FA4168F3328C2D33E192E7CA5D08F
5D4533F044A866D41&vpc_SecureHashType=SHA256
```

The Payment Server also includes the vpc_SecureHash in the Transaction Response so you can check the integrity of the receipt data. You do this by calculating the secure hash using the above method, then comparing your calculation with the value you received from TNS. If the values match, then you can be assured that we received the data you sent, and you received the data we sent.

> **Note**: Non-VPC fields (fields that do not begin with "vpc_") are returned ONLY for 3-Party integrations. In the Transaction Response,
>
> - the values for these fields cannot exceed 255 characters
> - the maximum number of fields returned are 5.
> - the maximum length of the response string in the URL cannot exceed 2048 characters.

Secure Hash Matching Error

Our Secure Hash method provides very good detection of attempts at fraud. However it is your responsibility to keep the key secret and to check the response. If the calculated and received values of the secure hash do not match, then you are at serious risk of eShoplifting. That is, providing your goods or service without being paid.

This could be due to:

- Fraud by your customer,
- Fraud by a man-in-the-middle attack (you are especially vulnerable to this if you do not use SSL between the customer's browser and your web site),
- Malicious corruption of the customer's web browser, or computer.

It is extremely unlikely that the reason was corruption by the network. There is only a one in one billion chance that a network packet will be corrupted and not corrected by the IP or TCP protocols.

Therefore you should take secure hash errors seriously, and when detected, take action that you think is appropriate to protect your business.

To simplify the calculation, the fields in the returned data in the Transaction Response are sorted in the order required for the Secure Hash calculation.

Store Secure Hash Secret Securely

You must keep your Secure Hash Secret stored securely. Do not store your secret within the source code of an ASP, JSP, or other website page as it is common for web server vulnerabilities to be discovered where source code of such pages can be viewed.

You should store your Secure Hash Secret in a secured database, or in a file that is not directly accessible by your web server and has suitable system security permissions.

You should change your Secure Hash Secret regularly in accordance with your company's security policy, and any time when you believe that its security may have been compromised.

You can change your Secure Hash secret in Merchant Administration in the Setup menu option on the Configuration Details page. For more information, please refer to your Merchant Administration User Guide.

Transaction Response Codes

The *vpc_TxnResponseCode* is a response code generated by the Payment Server that shows whether the transaction was successful. This response code can also be used to detect an error.

Any response code other than '0' is a declined/failed transaction. If the transaction is an error condition it will be contained in the vpc_Message field.

The response codes generated by the Payment Server are:

Code	Description
?	Response Unknown
0	Transaction Successful
1	Transaction could not be processed
2	Transaction Declined - Contact Issuing Bank
3	Transaction Declined - No Reply from Bank
4	Transaction Declined - Expired Card
5	Transaction Declined - Insufficient funds
6	Transaction Declined - Bank system error
7	Payment Server Processing Error - Typically caused by invalid input data such as an invalid credit card number. Processing errors can also occur.
8	Transaction Declined - Transaction Type Not Supported
9	Bank Declined Transaction (Do not contact Bank)
A	Transaction Aborted
B	Transaction Blocked - Returned when: ▪ the Verification Security Level has a value of '07'. ▪ the merchant has 3-D Secure Blocking enabled ▪ the overall risk assessment result returns a "Reject" or "System Reject".
C	Transaction Cancelled
D	Deferred Transaction
E	Transaction Declined - Refer to card issuer
F	3D Secure Authentication Failed
I	Card Security Code Failed
L	Shopping Transaction Locked (This indicates that there is another transaction taking place using the same shopping transaction number)
M	Transaction Submitted (the transaction has been directed to the acquirer but the Payment Server has not yet received it to complete the transaction)
N	Cardholder is not enrolled in 3D Secure (Authentication Only)
P	Transaction is Pending
R	Retry Limits Exceeded, Transaction Not Processed
T	Address Verification Failed

U	Card Security Code Failed
V	Address Verification and Card Security Code Failed

Address Verification Service (AVS) Response Codes

A security feature used for card not present transactions that compares the address entered by the cardholder with the records held in the card issuer's database. Once the transaction is successfully processed and authorized, the card issuer returns an address verification result code (AVS result code) in its authorization response message verifying the level of accuracy that matched the card billing address. These result codes are mapped to the AVS result codes returned by the Payment Server.

The AVS result codes returned by the Payment Server are:

Code	Description
X	Exact match – address and 9 digit ZIP/postal code
Y	Exact match – address and 5 digit ZIP/postal code
W	9 digit ZIP/postal code matched, Address not Matched
S	Service currently not supported.
G	International transaction, address information unavailable.
A	Address match only
C	Street Address and Postal Code not verified for International Transaction due to incompatible formats.
I	Visa Only. Address information not verified for international transaction.
Z	5 digit ZIP/postal code matched, Address not Matched
R	Issuer system is unavailable. Retry.
U	Address unavailable, no data from Issuer.
N	Address and ZIP/postal code not matched
E	Not a mailphone order.
0	No AVS requested. (Used by VisaII.)
B	Street Address match for international transaction. Postal Code not verified due to incompatible formats.
D	Street Address and postal code match for international transaction.
M	Street Address and postal code match for international transaction.
P	Postal Codes match for international transaction but street address not verified due to incompatible formats.
K	Card holder name only matches.
F	Street address and postal code match. Applies to U.K. only.

The AVS result codes for AMEX regions are:

AMEX AVS/AAV Response Code	TNS AVS/AAV Response Code	Code Description
Y	Y – exact match with 5 digit ZIP	Yes, Billing Address and Postal Code are both correct.
N	N – no match	No, Billing Address and Postal Code are both incorrect.

AMEX AVS/AAV Response Code	TNS AVS/AAV Response Code	Code Description
A	A – address match only	Billing Address only correct.
Z	Z – 5 digit ZIP match only	Billing Postal Code only correct.
U	U – address unavailable, no data from issuer	Information unavailable.
S	S – service not supported	SE not allowed AAV function.
R	R – issuer system unavailable	System unavailable; retry.
L	Y – exact match with 5 digit ZIP	CM Name and Billing Postal Code match.
M	Z – 5digit ZIP match only	CM Name, Billing Address and Postal Code match.
O	A – address match only	CM Name and Billing Address match.
K	K – card holder name only	CM Name matches.
G	A – address match only	CM and Alternate Ship-to information verified — Guaranteed
C	A – address match only	CM Billing and Ship-to information verified — No authorization; not guaranteed.

Card Security Code (CSC) Response Code

The Card Security Code (CSC) is a 3 or 4 digit numeric identifier printed on the signature panel of the card. For example, MasterCard and Visa use a 3 digit CSC and American Express has a 4 digit CSC.

It is a security feature used for card not present transactions that compares the Card Security Code entered by the cardholder with the records held in the card issuer's database. Once the transaction is successfully processed and authorized, the card issuer returns a result code (CSC result code) in its receipt response message, verifying the level of match the card issuer encountered.

The CSC result code in order of severity from highest (M) to lowest (N) are:

Code	Description	Level of Match
M	Valid or matched CSC	Highest
S	Merchant indicates CSC not present on card	
P	CSC Not Processed	
U	Card issuer is not registered and/or certified	
N	Code invalid or not matched	Lowest

External Payment Selection (EPS)

vpc_Gateway Field and Values

The vpc_Gateway field is used in External Payment Selection and determines what type of transaction is being performed. The field is case sensitive, and must comply with the following valid gateways in the Payment Server:

Code	Description
ssl	Specifies the gateway for all standard 3-Party transactions.
threeDSecure	Specifies the gateway for a 3-D Secure Mode 3a - 3-Party Style Authentication Only transaction.

Input 'vpc_Card' Field and Values

The vpc_Card field is used in External Payment Selection to select the card type that is to be used for the transaction.

The field is case sensitive, and must comply with each of the card types valid in the Payment Server. Please check with your Payment Provider as to which cards you can use.

The card Field values are:

Code	Description
Amex	American Express Credit Card
AmexPurchaseCard	American Express Corporate Purchase Card
Bankcard	Bankcard Credit Card
Dinersclub	Diners Club Credit Card
GAPcard	GAP Inc, Card
JCB	JCB Credit Card
Loyalty	Loyalty Card
Maestro	Maestro Debit Card
Mastercard	MasterCard Credit Card
Mondex	Mondex Card
PrivateLabelCard	Private Label Card
SafeDebit	SafeDebit Card
Solo	SOLO Credit Card
Style	Style Credit Card
Switch	Switch Credit Card

Code	Description
VisaDebit	Visa Debit Card
Visa	Visa Credit Card
VisaPurchaseCard	Visa Corporate Purchase Card

To check these values, open the 3-Party card selection page in a browser, and move the cursor over each card logo. The vpc_Gateway and vpc_Card values is displayed in the status bar at he bottom of the browser.

PayPal Acquirer

Enabling your merchant profile for the PayPal acquirer allows you to route transactions from your website to PayPal, if you choose PayPal as your payment method to perform the transaction. The transactions are routed through the 3-Party gateway, however; Payment Server pages are not displayed to the cardholder. For more information on the PayPal payment flows, see *3-Party Payments Using PayPal* in the *Virtual Payment Client Integration Guide*.

If you wish to use the PayPal functionality, you must have the following privileges enabled in your merchant profile.

Note: The Merchant Operator privileges are required to perform transactions through Merchant Administration.

Privilege	Merchant	Merchant Operator	AMA Operator
Virtual Payment Client	Yes	NA	NA
3-Party	Yes	NA	NA
External Pay Select	Yes	NA	NA
Advanced Merchant Administration	Yes	NA	Yes
Perform Voids	NA	Yes	Yes
Perform Captures	NA	Yes	Yes
Perform Refunds	NA	Yes	Yes

In some cases, PayPal may return a pending transaction response to the Payment Server (response code "P - Pending") and this is returned to the Merchant. PayPal will alert the Payment Server using a service called Instant Payment Notification (IPN) when the status of pending transaction is updated by PayPal. Based on the success or failure of the transaction, the Payment Server accordingly updates the response code. The merchant may use queryDR to retrieve the updated response code for the transaction.

PayPal Fields

These fields apply only to transactions that are routed through the PayPal acquirer. To perform a PayPal transaction, you must set vpc_Gateway to "**ssl**" and vpc_PaymentMethod to "**PAYPAL**". For more information, see ***External Payment Selection (EPS) Fields.*** on page 37

A PayPal transaction supports the following fields detailed in ***Acquirer Dependent Fields*** on page 39 section.

- vpc_ShipToAddressFromProvider
- vpc_CustomerIdFromProvider
- vpc_AcqResponseText
- vpc_ShipTo_FullName
- vpc_ShipTo_Street01
- vpc_ShipTo_City
- vpc_ShipTo_StateProv
- vpc_ShipTo_PostCode
- vpc_ShipTo_Country
- vpc_PaymentMethod

Note: Applies only to 3-party transactions.

PayPal Acquirer Response Code Mapping

This table shows the mapping between the Acquirer Response text returned by the PayPal acquirer to the TNS Response Code.

vpc_AcqResponseText	vpc_TxnResponseCode	Notes
Success : Expired	2 - Declined	-
Success : Failed	2 - Declined	-
Success : Reversal	2 - Declined	The transaction has been reversed. Most likely due to a failure with the Payment Review feature (PayPal risk checking)
Success : In-Progress	P - Pending	-
Success : Pending : Authorisation	0 - Approved	-
Success : Pending : XXX where **XXX** stands for all other values including PayPal risk checks, and account configuration issues.	P - Pending	Indicates the Payment Server has send a message and received a response, but is unable to confirm the transaction result.
Success : Processed	0 - Approved	-
SuccessWithWarning	P - Pending	-
Failure FailureWithWarning Warning	8 - Transaction Declined - Invalid request data received by Acquirer	The vpc_Message field includes the PayPal error message as follows: Exxxx: PayPal Acquirer Error: + PayPal Error Code + PayPal Long Message

3-D Secure Status Codes

All authentication transactions use a vpc_VerStatus response code value to show whether the card authentication was successful or not. The vpc_VerStatus response code values are:

Value	Description
Y	Success - The cardholder was successfully authenticated.
M	Success - The cardholder is not enrolled, but their card issuer attempted processing.
E	Not Enrolled - The cardholder is not enrolled.
F	Failed - An error exists in the request format from the Merchant.
N	Failed - Verification Failed.
S	Failed - The signature on the response received from the Issuer could not be validated. This should be considered a failure.
P	Failed - Error receiving input from Issuer.
I	Failed - Internal Error.
U	Undetermined - The verification was unable to be completed. This can be caused by network or system failures.
T	Undetermined - The cardholder session timed out and the cardholder's browser never returned from the Issuer site.
A	Undetermined - Authentication of Merchant ID and Password to the Directory Failed.
D	Undetermined - Error communicating with the Directory Server.
C	Undetermined - Card Type not supported.

The following vpc_VerStatus response codes are returned if "Use new 3DS response codes for VPC/PC" is enabled for the merchant profile.

Value	Description
Y	Success - The cardholder was successfully authenticated.
M	Success - The cardholder is not enrolled, but their card issuer attempted processing.
E	Undetermined - The Directory Server returned an Enrollment Status of "N" WITHOUT an Invalid Request element. This may indicate that the card cannot use 3DS.
F	Failed - An error exists in the request format from the Merchant.
N	Failed - Verification Failed.
S	Failed - The signature on the response received from the Issuer could not be validated. This should be considered a failure.
P	Failed - Error receiving input from Issuer.
I	Failed - Internal Error.

Value	Description
T	Undetermined - The cardholder session timed out and the cardholder’s browser never returned from the Issuer site.
A	Undetermined - Authentication of Merchant ID and Password to the Directory Failed.
D	Undetermined - Error communicating with the Directory Server.
C	Undetermined - Card Type not supported.
Z	Undetermined - The Directory Server returned an Enrollment Status of "N" WITH an Invalid Request element. The Invalid Request indicates that the Directory Server rejected the contents of at least one field in the request, i.e., the request was invalid.
B	Undetermined - The Directory Server returned an Enrollment Status of "U" WITHOUT an Invalid Request element.
V	Undetermined - The Directory Server returned an Enrollment Status of "U" WITH an Invalid Request element.
W	Undetermined - Unable to parse VERes received from the Directory Server.
X	Undetermined - The Access Control Server returned an Enrollment Status of "U".

Card Type Field Codes

The Card Type Field is a two character field that is returned in the receipt and identifies the card type that was used for this transaction.

Not all of these cards are available for all Payment Providers. Please check with your Payment Provider as to which cards you can use.

The Card Type Field values are:

Code	Description
AE	American Express
AP	American Express Corporate Purchase Card
BC	Bankcard
XC	Banamex Costco
DC	Diners Club
DS	Discover
FC	FarmersCard
JC	JCB Card
LS	Laser
SR	Soriana
MS	Maestro Card
MC	Mastercard
MP	Mastercard Purchase Card
PL	Private Label Card
QC	Q Card
SO	SOLO Card
ST	STYLE Card
TR	True Rewards Card
UA	UATP
VC	Visa Card
VD	Visa Debit Card
VP	Visa Corporate Purchase Card

Authorisation Response Data

Authorisation response data is additional data returned by the issuer during the authorisation process of a transaction. This data should be included in capture requests processed through an external system where applicable. When captures are processed through the Payment Server, this data is automatically included with the capture request as needed.

You can control the receipt of authorisation response data in the Transaction Response using the field vpc_ReturnAuthResponseData in the Transaction Request for both authorisation and purchase transactions. The received response data varies based on the card schemes, as shown below.

Note: A tick (✓) indicates the field is returned for that card scheme.

Authorisation Response Data	Visa	MasterCard	American Express	Discover
vpc_ReturnACI	✓	✗	✗	✗
vpc_TransactionIdentifier	✓	✓	✓	✓
vpc_CommercialCardIndicator	✓	✓	✗	✗
vpc_CardLevelIndicator	✓	✗	✗	✗
vpc_FinancialNetworkCode	✗	✓	✗	✗
vpc_MarketSpecificData	✓	✗	✗	✗

The Commercial Card field, vpc_CommercialCard, generated by the Payment Server, indicates if the card was identified by the issuer as a commercial card, based on the response returned from the issuer in the Commercial Card Indicator field, vpc_CommercialCardIndicator, as shown below.

vpc_CommercialCardIndicator		**vpc_CommercialCard**	
Code	**Description**	**Code**	**Description**
0 (zero)	Decline or not a Commercial Card	N	Not a Commercial Card
B	Business Card	Y	Commercial Card
R	Corporate Card	Y	Commercial Card
S	Purchasing Card	Y	Commercial Card
1	Consumer Card	N	Not a Commercial Card
2	Commercial Card	Y	Commercial Card

3	Both	U	Undetermined
Other	Undefined	U	Undetermined

Note: Codes 1-3 are returned only for MasterCard cards. Codes 0-S are returned for Visa cards.

Corporate Purchase Card Level 3 XML References

CPC Level III XML Document Type Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT CommodityCode (#PCDATA)>
<!ELEMENT CustomerRefNumber (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT DestinationZipCode (#PCDATA)>
<!ELEMENT Discount (DiscountAmount?, DiscountRate?)>
<!ELEMENT DiscountAmount (#PCDATA)>
<!ELEMENT DiscountRate (#PCDATA)>
<!ELEMENT ExtendedAmount (#PCDATA)>
<!ATTLIST ExtendedAmount   includesTax (Y | N) #REQUIRED>
<!ELEMENT Invoice (InvoiceSummary?, InvoiceDetail*)>
<!ELEMENT InvoiceDetail (InvoiceDetailNum, ProductCode?, Description?, Quantity?,
UnitOfMeasure?, ExtendedAmount?, UnitCost?, Discount?, Tax?, CommodityCode?)>
<!ELEMENT InvoiceDetailNum (#PCDATA)>
<!ELEMENT InvoiceSummary (CommodityCode?, CustomerRefNumber?, PurchaseIdentifier?,
LocalTaxAmount?, LocalTaxFlag?, DestinationZipCode?)>
<!ELEMENT LocalTaxAmount (#PCDATA)>
<!ELEMENT LocalTaxFlag (#PCDATA) >
<!ELEMENT LocalTaxRate (#PCDATA)>
<!ELEMENT ProductCode (#PCDATA)>
<!ELEMENT PurchaseIdentifier (#PCDATA)>
<!ELEMENT Quantity (#PCDATA)>
<!ELEMENT Tax (LocalTaxAmount?, LocalTaxRate?)>
<!ELEMENT UnitCost (#PCDATA)>
<!ELEMENT UnitOfMeasure (#PCDATA)>
```

CPC Level III XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="12DigitIntegerType">
    <xs:restriction base="xs:positiveInteger">
      <xs:totalDigits value="12"/>
      <xs:fractionDigits value="0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="YesNoType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Y"/>
      <xs:enumeration value="N"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="CommodityCode">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="40"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="CustomerRefNumber">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="17"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Description">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="40"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="DestinationZipCode">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="9"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Discount">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="DiscountAmount" minOccurs="0"/>
        <xs:element ref="DiscountRate" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DiscountAmount">
    <xs:simpleType>
      <xs:restriction base="12DigitIntegerType"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="DiscountRate">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        <xs:fractionDigits value="5"/>
        <xs:totalDigits value="12"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="ExtendedAmount">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="12DigitIntegerType">
          <xs:attribute name="includesTax" type="YesNoType" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Invoice">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="InvoiceSummary" minOccurs="0"/>
        <xs:element ref="InvoiceDetail" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
    <xs:unique name="NoDuplicateInvoiceDetailNums">
      <xs:selector xpath="InvoiceDetail"/>
      <xs:field xpath="InvoiceDetailNum"/>
    </xs:unique>
  </xs:element>
  <xs:element name="InvoiceDetail">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="InvoiceDetailNum"/>
        <xs:element ref="ProductCode" minOccurs="0"/>
        <xs:element ref="Description" minOccurs="0"/>
        <xs:element ref="Quantity" minOccurs="0"/>
        <xs:element ref="UnitOfMeasure" minOccurs="0"/>
        <xs:element ref="ExtendedAmount" minOccurs="0"/>
        <xs:element ref="UnitCost" minOccurs="0"/>
        <xs:element ref="Discount" minOccurs="0"/>
        <xs:element ref="Tax" minOccurs="0"/>
        <xs:element ref="CommodityCode" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="InvoiceDetailNum">
    <xs:simpleType>
      <xs:restriction base="12DigitIntegerType"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="InvoiceSummary">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="CommodityCode" minOccurs="0"/>
        <xs:element ref="CustomerRefNumber" minOccurs="0"/>
        <xs:element ref="PurchaseIdentifier" minOccurs="0"/>
        <xs:element ref="LocalTaxAmount" minOccurs="0"/>
        <xs:element ref="LocalTaxFlag" minOccurs="0"/>
        <xs:element ref="DestinationZipCode" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="LocalTaxAmount">
    <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
        <xs:totalDigits value="10"/>
        <xs:fractionDigits value="0"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="LocalTaxFlag">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="LocalTaxRate">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        <xs:totalDigits value="12"/>
        <xs:fractionDigits value="2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="ProductCode">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="40"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="PurchaseIdentifier">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="39"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Quantity">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        <xs:fractionDigits value="5"/>
        <xs:totalDigits value="12"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Tax">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="LocalTaxAmount" minOccurs="0"/>
        <xs:element ref="LocalTaxRate" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UnitCost">
    <xs:simpleType>
      <xs:restriction base="12DigitIntegerType"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="UnitOfMeasure">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="3"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

CPC Level III Example XML

```
<!DOCTYPE Invoice PUBLIC "-//Dialect Solutions/CPC Invoice 1.0//EN"
"http://www.dialectsolutions.com/DTDs/cpcInvoice.dtd">
<Invoice>
  <InvoiceSummary>
    <CommodityCode>Summary Commodity Code</CommodityCode>
    <CustomerRefNumber>Customer Ref Num</CustomerRefNumber>
    <LocalTaxAmount>66</LocalTaxAmount>
    <DestinationZipCode>12456</DestinationZipCode>
  </InvoiceSummary>
  <InvoiceDetail>
    <InvoiceDetailNum>1</InvoiceDetailNum>
    <ProductCode>Product Code 1</ProductCode>
    <Description>Description 1</Description>
    <Quantity>2.678</Quantity>
    <UnitOfMeasure>KG</UnitOfMeasure>
    <ExtendedAmount includesTax="N">440</ExtendedAmount>
    <UnitCost>220</UnitCost>
    <Discount>
      <DiscountAmount>22</DiscountAmount>
      <DiscountRate>4.76</DiscountRate>
    </Discount>
    <Tax>
      <LocalTaxAmount>44</LocalTaxAmount>
      <LocalTaxRate>10.0</LocalTaxRate>
    </Tax>
    <CommodityCode>Commodity Code 1</CommodityCode>
  </InvoiceDetail>
  <InvoiceDetail>
    <InvoiceDetailNum>2</InvoiceDetailNum>
      <ProductCode>Product Code 2</ProductCode>
      <Description>Description 2</Description>
      <Quantity>4.99</Quantity>
      <UnitOfMeasure>MG</UnitOfMeasure>
      <ExtendedAmount includesTax="Y">986</ExtendedAmount>
      <UnitCost>220</UnitCost>
      <Discount>
        <DiscountAmount>44</DiscountAmount>
        <DiscountRate>4.76</DiscountRate>
      </Discount>
      <Tax>
        <LocalTaxAmount>88</LocalTaxAmount>
        <LocalTaxRate>10.0</LocalTaxRate>
      </Tax>
    <CommodityCode>Commodity Code 2</CommodityCode>
  </InvoiceDetail>
</Invoice>
```

Error Codes

In an unsuccessful transaction with a vpc_TxnResponseCode of “7”, an error description may be contained in the field *vpc_Message* to describe the reason for the error.

The format of the error message is:

E*<error number>*-*<Date/Time Stamp MMDDHHMM>*: *<error description>*

For example: Where the error code is “5431” and the error description is “Invalid Field : CardNum”, the full error message returned is;

“*E5431-08131458: Invalid Field : CardNum*”

The common errors that a merchant may encounter are listed in the table below followed by a complete list of error codes that may be returned.

Error Codes and Their Descriptions for the Most Commonly Encountered Errors

Error Number	Description
5001	Invalid Digital Order
5004	Invalid Digital Order: invalid session ID
5005	Invalid Digital Order: invalid Merchant Id
5006	Invalid Digital Order: invalid purchase amount
5007	Invalid Digital Order: invalid locale
5050	Invalid Permission
5061	Unsupported payment method
5065	Runtime exception
5121	Try to access an invalid key file
5134	RSA Decrypt Failed
5135	RSA Encrypt Failed
5231	Retrieved Digital Receipt Error
5423	Bad User Name or Password
5425	Invalid Recurring Transaction Number
5426	Invalid Permission
5433	Invalid Permission
5435	Max No of Deferred Payment reached
5436	Invalid recurring transaction number

The complete list of Error Codes and their descriptions are:

Error Number	Description
5000	Undefined error
5001	Invalid Digital Order
5002	Invalid Digital Order: not enough fields
5003	Invalid Digital Order: too many fields
5004	Invalid Digital Order: invalid session ID
5005	Invalid Digital Order: invalid Merchant Id
5006	Invalid Digital Order: invalid purchase amount
5007	Invalid Digital Order: invalid locale
5008	Invalid Digital Order: outdated version
5009	Invalid Digital Order: bad or too many Transaction Request parameters. It could be one of the following: ▪ Invalid Digital Order: Invalid PAN Entry Mode ▪ Invalid Digital Order: Invalid PIN Entry Capability ▪ Bad Credit Payment Type ▪ Bad Account Balance Type ▪ Unsupported Transaction Type ▪ Invalid Digital Order: Invalid Payment Method ▪ Invalid Digital Order: Invalid PIN field ▪ Invalid Digital Order: Invalid KSN field ▪ Invalid Digital Order: Invalid STAN field ▪ Invalid Digital Order: Invalid PhysicalTerminalId field ▪ Invalid Digital Order: Invalid POSEntryMode field ▪ PIN Entry Capability Terminal Cannot Accept PIN ▪ PIN Entry Capability Terminal PIN pad down ▪ Authorisation Code must be provided ▪ Authorisation Code must be numeric and 1 to 6 characters in length

Error Number	Description
5010	Bad DCC Base Amount
5011	Bad DCC Base Currency
5012	Bad DCC Exchange Rate
5013	Bad DCC Offer State
5014	DCC Offer State Unsupported
5015	Missing or Invalid Currency
5016	Missing or Invalid Merchant Transaction Reference
5020	Invalid Digital Receipt
5021	Invalid Digital Receipt: not enough fields
5022	Invalid Digital Receipt: too many fields
5023	Invalid Digital Receipt: invalid session ID
5024	Invalid Digital Receipt: invalid Merchant Id
5025	Invalid Digital Receipt: invalid purchase amount
5026	Invalid Digital Receipt: invalid locale
5027	Error in generating Digital Receipt ID
5028	Invalid Digital Receipt Delivery URL
5029	Invalid Digital Receipt Delivery IO
5030	Invalid Transaction log string
5031	Invalid Transaction log string: not enough fields
5032	Invalid Transaction log string: too many fields
5033	Invalid Transaction log string: invalid purchase amount
5034	Invalid Transaction log string: invalid locale
5035	Transaction Log File error
5040	Invalid QsiFinTrans message
5041	Unsupported acquirer
5042	Unsupported transport
5043	Unsupported message format
5044	Invalid Merchant transaction mode
5045	Unsupported transaction counter
5046	SecureCGIParam verification of digital signature failed
5047	Failed to read a QsiSigner object back from a serialized file!
5048	Failed to create a DCOM object
5049	Receipt is invalid.
5050	Invalid Permission
5051	Unsatisfied DLL link error

Error Number	Description
5052	Invalid Merchant Id
5053	Transmission error from QSIFinTrans
5054	Parser error
5055	Acquirer Response Error
5056	Trace file I/O error
5057	Invalid cookie
5058	RMI exception
5059	Invalid session
5060	Invalid locale
5061	Unsupported payment method
5065	Runtime exception
5066	Bad parameter name or value
5070	File backup error
5071	File save error
5072	File IO error
5073	File not found error
5074	File not found
5080	SQL Error
5081	SQL Error : Cannot locate the database
5082	SQL Error : Cannot connect to the database
5083	SQL Error : Incorrect row count
5084	SQL Error : Invalid value format
5085	SQL Error : Bad line count
5086	Duplicate primary agent
5087	Unknown database type
5090	Illegal user name
5091	Illegal password error
5101	Could not create and load the specified KeyStore object. If you are using a QSIDB KeyStore the database connection may have failed
5103	Could not create the specified javax.crypto.Cipher object. You may not have a provider installed to create this type of Cipher object or the Cipher object that is specified in your config file is incorrect
5104	Error in call to javax.crypto.Cipher.doFinal. Either the input was too large or the padding was bad
5106	The Message type specified is not supported. Check the com.qsipayments.technology.security.MessageCrypto.properties file to ensure that the MsgType is valid
5108	The message received has a bad format

Error Number	Description
5109	Error verifying signature
5110	Error creating a signature
5161	Customer Reference too long
5175	Card track data exceeded the allowed lengths
5120	Unable to generate new keys
5121	Try to access an invalid key file
5122	Not able to store the security keys
5122	Not able to store the security keys
5123	Not able to retrieve the security keys
5124	Encryption format invalid for Digital Order
5125	Encryption signature invalid for Digital Order
5126	Invalid transaction mode
5127	Unable to find user keys
5128	Bad key Id
5129	Credit Card No Decryption failed
5130	Credit Card Encryption failed
5131	Problem with Crypto Algorithm
5132	Key used is invalid
5133	Signature Key used is invalid
5134	RSA Decrypt Failed
5135	RSA Encrypt Failed
5136	The keys stored in the keyfile given to SecureCGIParam was corrupt or one of the keys is invalid
5137	The private key stored in the keyfile given to SecureCGIParam was corrupt or one of the keys is invalid
5138	The public key stored in the keyfile given to SecureCGIParam was corrupt or one of the keys is invalid
5140	Invalid Acquirer
5141	Generic error for a financial transaction
5142	Generic reconciliation error for a transaction
5143	Transaction counter exceeds predefined value
5144	Generic terminal pooling error
5145	Generic terminal error
5146	Terminal near full
5147	Terminal Full
5148	Attempted to call a method that required a reconciliation to be in progress but this was not the case
5150	Invalid credit card: incorrect issue number length

Error Number	Description
5151	Invalid Credit Card Specifications
5152	Invalid Credit Card information contained in the database
5153	Invalid Card Number Length
5154	Invalid Card Number
5155	Invalid Card Number Prefix
5156	Invalid Card Number Check Digit
5157	Invalid Card Expiry Date
5158	Invalid Card Expiry Date Length
5162	Invalid Card Initialisation file
5166	Invalid Credit Card: incorrect secure code number length
5170	Unable to delete terminal
5171	Unable to create terminal
5161	Customer Reference too long
5175	Card track data exceeded the allowed lengths
5176	Bad Card Track, invalid card track sentinels
5185	Invalid Acknowledgement
5200	Payment Client Creation Failed
5201	Creating Digital Order Failed
5202	Creating Digital Receipt Failed
5204	Executing Administration Capture Failed
5205	Executing Administration Refund Failed
5206	Executing Administration Void Capture Failed
5207	Executing Administration Void Refund Failed
5208	Executing Administration Financial Transaction History Failed
5209	Executing Administration Shopping Transaction History Failed
5210	PaymentClient Access to QueryDR Denied
5220	Executing Administration Reconciliation Failed
5221	Executing Administration Reconciliation Item Detail Failed
5222	Executing Administration Reconciliation History Failed
5230	Retrieving Digital Receipt Failed
5231	Retrieved Digital Receipt Error
5232	Digital Order Command Error
5233	Digital Order Internal Error
5234	MOTO Internal Error
5235	Digital Receipt Internal Error

Error Number	Description
5336	Administration Internal Error
5400	Digital Order is null
5401	Null Parameter
5402	Command Missing
5403	Digital Order is null
5410	Unknown Field
5411	Unknown Administration Method
5412	Invalid Field
5413	Missing Field
5414	Capture Error
5415	Refund Error
5416	VoidCapture Error
5417	VoidRefund Error
5418	Financial Transaction History Error
5419	Shopping Transaction History Error
5420	Reconciliation Error
5421	Reconciliation Detail Error
5422	Reconciliation History Error
5423	Bad User Name or Password
5424	Administration Internal Error
5425	Invalid Recurring Transaction Number
5426	Invalid Permission
5427	Purchase Error
5428	VoidPurchase Error
5429	QueryDR Error
5430	Missing Field
5431	Invalid Field Digital.TRANS_NO must be provided to indicate which existing order this transaction is to be performed against
5432	Internal Error
5433	Invalid Permission
5434	Deferred Payment service currently unavailable
5435	Max No of Deferred Payment reached
5436	Invalid recurring transaction number
5450	DirectPaymentSend: Null digital order
5451	DirectPaymentSend: Internal error

Error Number	Description
5500	Error in card detail
5501	Errors exists in card details
5600	Transaction retry count exceeded
5601	Instantiation of AcquirerController for this transaction failed.
5602	An I/O error occurred
5603	Could not get a valid terminal
5604	Unable to create the ProtocolReconciliationController for the protocol
5661	Illegal Acquirer Object Exception
5670	Message Exception
5671	Malformed Message Exception
5672	Illegal Message Object Exception
5680	Transport Exception
5681	Transport type not found
5682	Transport connection error
5683	Transport IO error
5684	Illegal Transport Object Exception
5690	Permanent Socket Transport connected
5691	Permanent Socket Transport JII class exception
5692	Permanent Socket Transport mismatched message received
5693	Permanent Socket Transport malformed message received
5694	Permanent Socket Transport unavailable
5695	Permanent Socket Transport disconnected
5696	The connection has been closed prematurely
5730	Host Socket unavailable
5750	Message header not identified
5751	Message length field was invalid
5752	Start of text marker (STX) not found where expected
5753	End of text marker (ETX) not found where expected
5754	Message checksum (LRC) did not match
5800	Init service started
5801	Init service stopped
5802	Invalid entry
5803	Duplicate entry
5804	Parse error
5805	Executing task

Error Number	Description
5806	Cannot execute task
5807	Terminating task
5808	Task killed
5809	Respawning task
5810	Cron service started
5811	Cron service stopped
5812	Parse error
5813	Invalid entry
5910	Null pointer caught
5911	URL Decode Exception occurred
5930	Invalid card type for excessive refunds
5931	Agent is not authorized to perform excessive refunds for this amount
5932	Too many excessive refunds apply to this shopping transaction already
5933	Merchant agent is not authorized to perform excessive refunds
5934	Merchant is not authorized to perform excessive refunds
5935	Merchant cannot perform excessive refunds due to its transaction type
6010	Bad format in Rulefile
6100	Invalid host name
7000	XML parser [Fatal Error]
7001	XML parser [Error]
7002	XML parser [Warning]
7003	XML Parameter is invalid
7004	XML Parameter had an invalid index. Check input .html file
7005	XML [Bad Provider Class]
7050	SleepTimer: Time value is not in a valid format (ignored this time value)
7100	No valid times and/or interval specified in StatementProcessing.properties file. Execution terminated
7101	Status file for this data file was never created – deleting
7102	Error loading Statement.properties file
7104	Can’t find file
7106	IOException thrown attempting to create or write to file
7107	Overwriting file
7108	SecurityException thrown when attempting to create output file
7109	Invalid Merchant Id. This Advice element will not be processed
7110	Can’t create file name from the given date string
7111	Duplicate Advice element found in input document and skipped. Check input document

Error Number	Description
7112	Invalid payment type specified. This file will be skipped
7113	Null directory: can’t create output file
7114	Validation of input file provided by host failed
7120	IOException thrown attempting to create or write to file
7121	IOException thrown while attempting to create a ZIP archive
7122	An inaccessible output directory was specified in the configuration file
7200	PRE Issue Id Error
7201	No Login User Object stored in session.
7202	Error Occurred while creating the merchant on the Payment Server.
7203	Logging out
7204	Error occurred while instantiating Payment.
7205	Error occurred while instantiating SSL Payment
7207	Error occurred while sending email
7208	Invalid Access. User is trying to access a page illegally.
7209	Invalid User Input.
7300	Error parsing meta data file
7301	Invalid field
7302	Field validator not present
7303	Validation of field failed
7304	Field not present in arbitrary data
7305	Mandatory field missing
7306	Date mask is invalid
7307	Error creating field validator
7308	Failed to update arbitrary data
7400	Invalid transaction type
7500	Record has changed since last read
8000	Invalid Local Tax Flag
8001	Local Tax Amount Equal to or Greater then Initial Transaction Amount
8002	Purchaser Postcode Too Long
8003	Invalid Local Tax Flag and Local Tax Flag Amount Combination
8004	Invalid Local Tax Amount
8015	Payment method must be EBT for a balance inquiry
8015	Invalid Digital Order: Invalid PaymentMethod
8016	Invalid Digital Order: Invalid PIN field
8017	Invalid Digital Order: Invalid KSN field

Error Number	Description
8019	Invalid Digital Order: Invalid PhysicalTerminalID field
8020	Invalid Digital Order: Invalid POSEntryMode field
8021	Invalid Digital Order: Invalid AdditionalAmount field
9000	Acquirer did not respond
9052	UNSUPPORTED_PAYMENT_PLAN; returned if Payment Plan is not configured for the selected Merchant Acquirer link. Used for system-level payment plans.
9053	UNSUPPORTED_CUSTOM_PAYMENT_PLAN; returned if the custom Payment Plan does not match custom plans for the selected Merchant Acquirer link.
9054	UNSUPPORTED_NUM_PAYMENTS; returned if the requested number of payments is not supported by the selected Payment Plan or Payment Plan/Custom Payment Plan combination.
9055	UNSUPPORTED_NUM_DEFERRALS; returned if the requested number of deferrals is not supported by the selected Payment Plan or Payment Plan/Custom Payment Plan combination.
9056	INVALID_PAYMENT_PLAN_REQUEST; returned if the request contained both Payment Plan and Custom Payment Plan when only one or the other is expected.
9150	Missing or Invalid Secure Hash
9151	Invalid Secure Hash Type, or Secure Hash Type not allowed for this merchant
9152	Missing or Invalid Access Code
9153	Request contains more than one instance of the same field [FieldName]
9154	General merchant configuration error preventing request from being processed
9200	Missing or Invalid Template Number

Index