Virtual Payment Client

Reference Guide

Version 20.2.2

For MIGS 20.2.2

Notices

Following are policies pertaining to proprietary rights and trademarks.

Proprietary Rights

Trademarks

Disclaimer

Summary of Changes, 26 June 2020

Description of Changes
Enhancements to the integration information on EMV 3DS (3DS2) functionality

Summary of Changes, 13 May 2020

Description of Changes
Added integration information on EMV 3DS (3DS2) functionality
Updated vpc_ReturnURL description for the HTTPS mandatory requirement

Summary of Changes, 19 April 2019

Description of Changes
Added integration information on card on file, cardholder-initiated and merchant-initiated transactions

Summary of Changes, 19 April 2019

Description of Changes
Added integration information for card scheme tokens

Summary of Changes, 25 January 2019

Description of Changes
Updated descriptions for vpc_3DSECI and vpc_VerSecurityLevel fields to include the note about 07 ECI value.

Summary of Changes, 22 August 2018

Description of Changes
Updated descriptions for vpc_3DSECI and vpc_VerSecurityLevel fields.

Summary of Changes, 6 October 2017

Description of Changes
Added vpc_OrderCertainty field for both 2-party and 3-party authorization transactions

Summary of Changes, 17 March 2017

Description of Changes
Removed vpc_OrderCertainty field for both 2-party and 3-party authorization transactions
Fixed description for the vpc_TransNo field for all subsequent transaction types

Summary of Changes, 22 August 2017

Description of Changes
Enhanced field descriptions for vpc_SecureHash, vpc_SecureHashType, and vpc_Card input fields

Summary of Changes, 8 September 2016

Description of Changes
Added vpc_OrderCertainty field for both 2-party and 3-party authorization transactions
Fixed description for the vpc_TransNo field for all subsequent transaction types

Contents

Preface 9

Audience 9
Where to Get Help....9

Introduction 11

How This Guide is Structured 11
Related Documents and Materials 12
Terminology....13

Basic Transaction Fields 15

Field Types 15
Input Requirements 16
2-Party Payment Model 16
3-Party Payment Model 16
Input Fields for Basic 2-Party Transactions 17
Input Fields for Basic 3-Party Transactions 21
Basic Output Fields 24

Supplementary Transaction Fields 29

Address Verification Service (AVS) Fields 29
Transaction Request Input Fields 30
Transaction Response Output Fields....30
Card Present Fields....32
Transaction Request Input Fields 32
Transaction Response Output Fields....35
Card Security Code (CSC) Field 35
Transaction Request Input Fields 35
Transaction Response Output Fields....36
External Payment Selection (EPS) Fields....37
Transaction Request Input Fields 37
Transaction Response Output Fields....38
MasterPass Fields....38
Transaction Request Input Fields 38
Transaction Response Output Fields....39
Merchant Transaction Source 40
Transaction Request Input Fields 41
Transaction Response Output Fields....41
Merchant Transaction Source Frequency....42
Transaction Request Input Fields 42
Transaction Response Output Fields....42
Enhanced Industry Data Fields 43
Transaction Request Input....43
Transaction Response Output 43
Referral Message Fields 44
Transaction Request Input Fields 44
Transaction Response Output Fields....44
Referral Processing Transaction Fields 45
Transaction Request Input Fields 45
Transaction Response Output Fields....46
Risk Management Fields....47
Transaction Request Input Fields 47
Transaction Response Output Fields....48
Bank Account Type Field 49

Transaction Request Input Fields 49
Transaction Response Output Fields....49
ANZ Bank Extended OrderInfo Field....50
Transaction Request Input Fields 50
Transaction Response Output Fields....50
CashAdvance 51
Transaction Response Output Fields....51
Verification Only 52
Transaction Request....52
Transaction Response 54
Credential on File Fields 57
Cardholder-initiated Transactions 57
Merchant-initiated Transactions....57
Transaction Request Input Fields 58
Transaction Response 59
Card Scheme Tokens....59
Transaction Request....59
Transaction Response 60
Payment Authentication 62
Key Benefits 62
3DS Authentication Versions 62
Prerequisites 63
Payment Server Integration Modes for 3DS Authentication 63
Mode 1 - Combined 3-Party Authentication & Payment Transaction (Payment Server collects card details) 68
Mode 2 - Combined 3-Party Authentication and Payment transaction (Merchant collects card details)....75
Mode 3a - 3-Party Authentication Only (Merchant collects card details)....78
Mode 3b - **2**-Party Pre-Authenticated Payment Transaction (Merchant supplies authentication details) 80
3-D Secure JavaScript API Integration 87
threeds2.js API Reference 88
URL 88
Functions....88
Callbacks....90
Error Codes....90
Explanation Values 91

Advanced Merchant Administration (AMA) Transactions 93

Basic Transaction Fields 95
Basic Input Fields - AMA Transaction....96
Basic Output Fields - AMA Transaction....98
AMA Capture Transaction....101
Transaction Request Input Fields 101
Transaction Response Output Fields....101
AMA Refund Transaction 103
Transaction Request Input Fields 103
Transaction Response Output Fields....103
AMA Void AuthorisationTransaction 105
Transaction Request Input Fields 105
Transaction Response Output Fields....105
AMA Void Capture Transaction....106
Transaction Request Input Fields 106
Transaction Response Output Fields....106
AMA Void Refund Transaction....108
Transaction Request Input Fields 108
Transaction Response Output Fields....108
AMA Void Purchase Transaction 110

Transaction Request Input Fields 110
Transaction Response Output Fields....110
AMA Standalone Capture Transaction....112
Transaction Request Input Fields 112
Transaction Response Output Fields....114
AMA Standalone Refund Transaction....115
Transaction Request Input Fields 115
Transaction Response Output Fields....117
AMA QueryDR....118
Transaction Request Input Fields 118
Transaction Response Output Fields....118

References - Virtual Payment Client 121

Generating a Secure Hash....121
Creating a SHA-256 HMAC Secure Hash 121
Merchant- Supplied Parameters 121
SHA-256 HMAC Calculation 122
Secure Hash Matching Error....123
Store Secure Hash Secret Securely 124
Transaction Response Codes....125
Address Verification Service (AVS) Response Codes....131
Card Security Code Response Code 132
External Payment Selection (EPS) 133
vpc_Gateway Field and Values 133
Input 'vpc_Card' Field and Values 133
3-D Secure Status Codes 135
Card Type Codes 137
Authorisation Response Data 138
Card Present Data....139
Error Codes 140
Error Codes and Their Descriptions for the Most Commonly Encountered Errors....140

Index 151

CHAPTER 1

Preface

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 Mastercard.

CHAPTER 2

Introduction

Mastercard's 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 Mastercard 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 Mastercard 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 32 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. It is required that you use HTTP POST 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 in the request body using HTTP POST. A failure to conform to this rule will result in a HTTP Response 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**'. Depending on the transaction mode configured for the merchant profile, an Authorization or Purchase transaction is performed.			
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

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. **Note**: This field is optional for Maestro card transactions. If you do not provide a value, the field defaults to 4912 (Dec 2049).			
Required	Numeric	4	1305
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

vpc_SecureHash			
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. **Note**: This field is required if “Enforce Secure Hash (2-party)” privilege is enabled on your merchant profile. For more details see ***Generating a Secure Hash*** on page 121 and remember to ***always store the Secure Hash secret securely*** on page 124. **Note:** The secure secret is provided by the Payment Provider.			
Optional	Alphanumeric	64	9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2
vpc_SecureHashType			
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 **Note**: This field is required if “Enforce Secure Hash (2-party)” privilege is enabled on your merchant profile. For more details see ***Generating a Secure Hash*** on page 121.			
Optional	Alphanumeric	6	SHA256
vpc_ReturnAuthResponseData			
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 ***Authorisation Response Code*** on page 138.			
Optional	Alpha	1	Y

vpc_OrderCertainty			
Indicates if you expect to capture the full order amount for which you are requesting authorization. Depending on your merchant profile configuration, you may be able to provide: ▪ **FINAL**: The full authorized amount is expected to be captured within the mandated time. The order will only be cancelled in exceptional circumstances (for example, the cardholder cancelled their purchase). ▪ **ESTIMATED**: The authorized amount is an estimate of the amount that will be captured. It is possible that the amount captured will be less, or might not be captured at all. If this field is not provided, the default order certainty level configured for you by your MSO will be used. If a default is not configured, the gateway default FINAL will be used. The value for this field in the transaction response indicates the value the gateway will send to the acquirer. **Note**: Applies only to authorization transactions. For a Pay transaction, the gateway default FINAL is used.			
Optional	Alpha	1, 24	ESTIMATED

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>/vpcpay

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 set to '**pay**'. Depending on the transaction mode configured for the merchant profile, an Authorization or Purchase transaction is performed.			
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

vpc_Locale			
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.			
Required	Alphanumeric	2,5	en

vpc_ReturnURL			
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. **Note**: The return URL must be a fully qualified URL starting with HTTPS://. It is required that the browser is redirected to a TLS secured page, which supports at a minimum TLS version 1.2.			
Required	Alphanumeric	1,255	https://merchants_site/receipt.asp
vpc_SecureHash			
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. **Note**: This field is not required if “May Omit Secure Hash (3-Party) is enabled on your merchant profile. For more details see ***Generating a Secure Hash*** on page 121 and remember to ***always store the Secure Hash secret securely*** on page 124. **Note:** The secure secret is provided by the Payment Provider.			
Required	Alphanumeric	64	9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2
vpc_SecureHashType			
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 **Note**: This field is not required if “May Omit Secure Hash (3-Party) is enabled on your merchant profile. For more details see ***Generating a Secure Hash*** on page 121.			
Required	Alphanumeric	6	SHA256
vpc_ReturnAuthResponseData			

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 ***Authorisation Response Code*** on page 138.			
Optional	Alpha	1	Y

vpc_OrderCertainty			
Indicates if you expect to capture the full order amount for which you are requesting authorization. Depending on your merchant profile configuration, you may be able to provide: ▪ **FINAL**: The full authorized amount is expected to be captured within the mandated time. The order will only be cancelled in exceptional circumstances (for example, the payer cancelled their purchase). ▪ **ESTIMATED**: The authorized amount is an estimate of the amount that will be captured. It is possible that the amount captured will be less, or might not be captured at all. If this field is not provided, the default order certainty level configured for you by your MSO will be used. If a default is not configured, the gateway default FINAL will be used. The value for this field in the transaction response indicates the value the gateway will send to the acquirer. **Note**: Applies only to authorization transactions. For a Pay transaction, the gateway default FINAL is used.			
Optional	Optional	Optional	Optional

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 0 (successful).			
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

A unique transaction ID 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 to identify the target transaction when performing subsequent transactions such as refund, capture and void.
This field is not returned for transactions that result in an error condition.

Output	Numeric	1,19	96841

vpc_ShopTransactionNo

A unique order number generated by the Payment Server for the transaction. All subsequent transactions you perform on this transaction will be assigned the same order number.

Output	Numeric	1,19	10712

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*** on page 137.
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 124.

Output	Alphanumeric	64	9FF46885DCA8563ACFC62058E0FC447BD2C033D505BD8202F681DCAD7CED4DD2

vpc_SecureHashType

The value of vpc_SecureHashType returned in the Transaction Response.

Input	Alphanumeric	6	SHA256

vpc_CardNum

The card number in 0.4 card masking format.
This field is only returned if *System-Captured Masked Card in Digital Receipt* privilege is enabled for the merchant processing the transaction. See *Merchant Manager User Guide.*

Note: Applies only to 3-party transactions.

Output	Alphanumeric Special	5	-1234
vpc_ReturnACI			
The ACI (Authorization Characteristics Indicator) returned by the issuer. For information, see ***Authorization Response Code*** on page 138. **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 138. **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 138. **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 138. **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 138. **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 138. **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 138. **Note**: This field is returned only if vpc_ReturnAuthResponseData was specified as "Y" in the Transaction Request.			
Output	Alphanumeric	0,1	A

vpc_OrderCertainty			
The certainty level on the authorized amount that will be captured for this transaction. This is the order certainty value the gateway will send to the acquirer. Valid values for this field are: ▪ **FINAL**: The full authorized amount is expected to be captured within the mandated time. The order will only be cancelled in exceptional circumstances (for example, the payer cancelled their purchase). ▪ **ESTIMATED**: The authorized amount is an estimate of the amount that will be captured. It is possible that the amount captured will be less, or might not be captured at all.			
Input	Alpha	1,24	ESTIMATED

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

Address Verification Service (AVS) 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_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

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. **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

Card Present Fields

Card present payments refer to transactions using a Point of Sale (POS) terminal. The terminal may read card data by:

- keying the card number
- swiping a magnetic stripe card
- inserting an EMV card
- NFC from a contactless card

The card data generated from the terminal is included in the Transaction Request with an Authorisation, Purchase, or Capture transaction. Card present functionality can only be performed as a 2-Party Authorisation/Purchase/Capture transaction.

For all card present transactions the Merchant Transaction Source (vpc_TxSource) must be set to **'CARDPRESENT'**.

For a magnetic stripe swipe, the card track data (vpc_CardTrack1 and vpc_CardTrack2) needs to contain the correct start and end sentinel characters and trailing longitudinal redundancy check (LRC) characters.

If the magnetic stripe data is not available, for example, if the card is defective, or the POS terminal was malfunctioning at the time, it is sufficient to set the merchant transaction source to '**CARDPRESENT**' and change the '*PAN Entry Mode*' and '*PIN Entry Mode'* values in *vpc_POSEntryMode* field to indicate that the card was sighted, but manually entered.

To be able to submit EMV transactions, merchants must have “May perform EMV transactions” privilege. Both contact and contactless EMV transactions are supported.

Note: Card Track 3 data is not supported.

Transaction Request Input Fields

Card Present 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_CardTrack1			
7 bit ASCII text representing the card track 1 data.			
Optional	Alphanumeric	2, 79	%B5123456789012346^MR JOHN R SMITH ^13051019681143300001 840 ?;
vpc_CardTrack2			

7 bit ASCII text representing the card track 2 data. The contents of vpc_CardTrack2 must match the PAN and expiry fields included in the Transaction Request.			
Optional	Alphanumeric	2,40	;5123456789012346=13051019681143384001?

vpc_POSEntryMode			
The first 2 characters define the actual PAN Entry Mode and the third character defines the PIN Entry Mode. **PAN ENTRY Mode** 01 - Manual Entry 02 - Magnetic stripe read, but full unaltered contents not provided 04 - OCR/MICR coding read 90 - Magnetic stripe read and full, unaltered contents provided 05 - PAN auto entry via chip 79 - Chip card at chip-capable terminal was unable to process transaction using data on the chip or magnetic stripe on the card-therefore, PAN entry via manual entry 80 - Chip card at chip-capable terminal was unable to process transaction using data on the chip therefore, the terminal defaulted to the magnetic stripe read for the PAN. This is referred to as fallback. 07 - Auto-entry via contactless magnetic chip 91 - Auto-entry via contactless magnetic strip **PIN Entry Mode** 0 - Unspecified or unknown 1 - Terminal has PIN entry capability 2 - Terminal does not have PIN entry capability (default) 8 - Terminal has PIN entry capability but PIN pad is not currently operative. See ***Card Present codes*** on page 139 for more information.			
Required	Numeric	3	052

vpc_CardSeqNum			
The card sequence number for transactions where the data is read through a chip on the EMV card.			
Optional	Numeric	3	133
vpc_EMVICCData			
Data read through a chip on the EMV card, base64 encoded.			
Required	Alphanumeric	1,340	QUJDMzQ1

vpc_TxSource			
The source of the transaction. This must be set to CARDPRESENT if the merchant's default transaction source has not been configured to CARDPRESENT.			
Optional	Alphanumeric	11	CARDPRESENT
vpc_TerminalAttended			
Specifies whether the terminal is attended by the merchant. Valid values are: Y - indicates that the terminal is attended. N - indicates that the terminal is unattended. U - indicates that the status is unknown or unspecified.			
Optional	Alphanumeric	1	Y
vpc_CardholderActivatedTerminal			

Specifies whether the terminal is activated by the cardholder. Valid values are: N - indicates that the terminal is not activated by the cardholder. SS - indicates that the terminal is self serviced.			
Optional	Numeric	1, 2	SS
vpc_TerminalInputCapability			
Indicates the input capability of the terminal. Valid values are: M Magnetic strip read (MSR) only (currently not supported) KM MSR and key entry (currently not supported) K Key entry only (currently not supported) CM MSR and chip CKM MSR, chip and key entry C Chip read only MX Contactless MSR CX Contactless chip			
Optional	Numeric	1, 5	MX
vpc_TerminalLocation			
Specifies the location of the terminal in relation to the premises of the card acceptor. Valid values are: P - indicates that the terminal is on the premises of the card acceptor. O - indicates that the terminal is off the premises of the card acceptor.			
Optional	Alphanumeric	1	P
vpc_POSTerminalName			
The name that you use to identify the Point Of Sale (POS) instance. This should uniquely identify one POS within your business. This field can be used for your search or reporting needs, and might be used by risk processing systems.			
Optional	Alphanumeric	1,8	S43_L12 (for Lane 12 in Shop 43) or Kiosk_76

Transaction Response Output Fields

Card Present Output Fields			
In addition to the standard output fields, the following optional fields are also returned in the Transaction Response for 2-Party transactions.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_EMVICCData			
The value of the vpc_EMVICCData input field returned in the Transaction Response.			
Output	Alphanumeric	1, 340	QUJDMzQ1

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(American Express) 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 merchant's application must send the *vpc_CardSecurityCode* value, if CSC is required.

You can enforce CSC on transaction sources using "Enforce CSC on Transaction Sources" merchant privilege, which then enforces collection of CSC for selected transaction sources. Note that CSC enforcement does not apply to:

- Card Present transactions
- transactions where the transaction frequency is Recurring or Installment.
- transactions with Maestro cards for a transaction source of Internet.

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(American Express) 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

vpc_CSCResultCode			
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 '**Unsupported**'.			
Output	Alpha	1,11	M

vpc_AcqCSCRespCode			
The result code generated by the card issuing institution in relation to the Card Security Code. This is only provided for ancillary information.			
Output	Alpha	1,11	M

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 “External Pay Select” 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			
Specifies the card type used in the 3-party transaction. The field is case sensitive, and must match one of the card types assigned to the merchant in their merchant profile. To check the card types available to you, 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. The possible values for the input field vpc_Card are shown in ***External Payment Selection (EPS)*** on page 133.			
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.			
Optional	Alpha	3,6	CREDIT

Transaction Response Output Fields

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

MasterPass Fields

MasterPass is a digital wallet that allows customers to store details of one or more credit cards in a secure server. The customer can also choose to store other details such as billing address and shipping address. If you are enabled for MasterPass, you can allow the Payment Server to launch the MasterPass lightbox where the customers can select their payment and shipping address details.

To offer MasterPass as an option, your merchant profile must be enabled and configured for the MasterPass service and the 3-Party pages.

Note: Applies to 3-party transactions only.

Transaction Request Input Fields

MasterPass 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_CardSource			
Indicates that the source of the card details is a digital wallet containing one or more credit cards. Use this field if you wish to launch the MasterPass lightbox directly from your website rather than allowing the customer to select the digital wallet on the Payment Server card selection page. Valid values for this field are: ▪ MASTERPASS **Note**: *vpc_CardSource* takes precedence over *vpc_Card* when processing transactions. For example, if *vpc_CardSource*=MASTERPASS and *vpc_Card*=SomeCard then *vpc_Card* is ignored and the customer is directly presented with the MasterPass lightbox.			
Optional	Alphanumeric	10	MASTERPASS

vpc_ReturnMasterPassResponseParameters			
Specifies whether the fields specific to MasterPass, *vpc_WalletIndicator* and *vpc_AVS_CardMemberName*, are returned in the transaction response. Valid values for this field are: Y - indicates that these fields are returned in the transaction response. N - indicates that these fields are not returned in the transaction response. This is the default value.			
Optional	Alpha	1	Y

Transaction Response Output Fields

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

vpc_WalletIndicator			
The identifier returned by MasterPass if the MasterPass digital wallet was used by the customer to provide the payment details for this transaction. **Note**: This field is returned in the transaction response only if *vpc_ReturnMasterPassResponseParameters*=Y.			
Output	Numeric	3	101

vpc_ShipTo_Street01			
The street name and number, or the Post Office Box details, of the address to which the current order is being shipped.			
Output	Alphanumeric	1,128	1136 John Street

vpc_ShipTo_City			
The city to which the current order is being shipped.			
Output	Alphanumeric	1,128	Seattle

vpc_ShipTo_StateProv			
The state or province to which the current order is being shipped.			
Output	Alphanumeric	0,128	WA

vpc_ShipTo_PostCode			
The post code or zip code of the address to where the current order is being shipped.			
Output	Alphanumeric	4,9	98111

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

vpc_ShipTo_LastName			
The last name or surname of the person to whom current order is being shipped.			
Output	Alphanumeric	0,1	Doe
vpc_ShipTo_FirstName			
The first name of the person to whom the current order is being shipped.			
Output	Alphanumeric	1,15	Jane
vpc_ShipTo_Phone			
The phone number of the contact person to whom the current order is being shipped.			
Output	Alpha	3	USA

vpc_AVS_CardMemberName			
The cardholder name collected at MasterPass. **Note**: This field is returned in the transaction response only if *vpc_ReturnMasterPassResponseParameters*=Y.			
Optional	Alphanumeric	1, 128	Alan Adam

Note 1: The shipping address fields are returned only if the merchant profile is configured to collect the customer's shipping address at MasterPass.

Note 2: If you provide a billing or shipping address in the 3-Party payments request and also allow MasterPass to collect a billing or a shipping address, then the address returned by MasterPass will completely override the address provided in the 3-Party payments request.

Note 3: If you are not enabled for the "May Use AVS" privilege and do not provide the billing address details in the 3-Party payments request then the billing address returned by MasterPass will neither be stored against the transaction, returned in the transaction response, nor sent to the acquirer.

Note 4: If MasterPass Online returns non-Latin-1 characters, then the non-Latin-1 characters will be converted to '?' characters and stored against the transaction only for the following fields.

vpc_AVS_CardMemberName
vpc_AVS_City
vpc_AVS_Country
vpc_AVS_StateProv
vpc_AVS_Street01
vpc_AVS_PostCode

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. **MERCHANT** - indicates that the transaction was initiated by the merchant based on an agreement with the cardholder. For example, a recurring payment, installment payment, or account top-up. **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 your Payment Provider.			
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.

Enhanced Industry Data Fields

Although Enhanced Industry Data functionality was originally designed for the travel industry, this functionality allows the merchant to enter any industry related data to be stored on the Payment Server for that transaction. It includes fields:

- Ticket Number — allows the merchant to submit airline ticket number in the Transaction Request, including Capture transactions. The previous ticket number is overwritten when a new ticket number is submitted and the Payment Server does not maintain an audit record of the changes. You can view the latest Ticket Number in the search results of a Transaction Search using the Merchant Administration portal on the Payment Server.
- Addendum Data — allows the merchant to include industry specific data in the Transaction Request. The data can include passenger names, ticket numbers, hotel bookings, etc. The addendum data is stored in the database, which may be used in creating reports external to the Payment Server.

Both Ticket number and Addendum Data are passed with the Transaction Request and stored on the Payment Server. The ticket number is passed to the financial institution as part of certain transactions.

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

Transaction Request Input

Enhanced Industry Data 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
vpc_AddendumData			
Extra information about the industry, for example, passenger names, ticket numbers, hotel bookings, etc., that is passed with the Transaction Request and stored on the Payment Server. **Prerequisite:** You must enable the privilege *May Include Addendum Data* to pass Addendum data in the Transaction Request.			
Note: Though AddendumData supports 4000 characters, ensure that the Transaction Request does not exceed 4000 characters due to browser redirect limitations in 3-party transactions.			
Optional	Alphanumeric Special	0, 4000	Scott Adam, VIP Client, Acme Hotel.

Transaction Response Output

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.

The Authorisation code the merchant is given on contacting the Payment Provider is input using a '**Referral Transaction** on page 45'.

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*** (see "Returned Response Codes" on page 125). 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 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			
Provide the value returned in the vpc_TransactionNo field for the referred initial transaction (authorization/purchase) that you wish to resubmit.			
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.

When you are configured to use Risk Management through Virtual Payment Client, transactions processed through the Virtual Payment Client will be assessed for risk, and the risk recommendation for each authorization and purchase will be returned in the Transaction Response. Orders that are flagged for review as a result of risk assessment may be reviewed for acceptance or rejection only through the Merchant Administration portal. You can view the risk assessment details in the search results of an Order Search using Merchant Administration.

Risk management is only applicable to the first transaction on the order, which may be an Authorization, Pay, or Verification Only. Risk assessment of other transactions such as Standalone Captures, Standalone Refunds, or Voids is not performed.

Note: As a prerequisite, merchants must be enabled for the *Internal Risk Rules* privilege.

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 authorization or purchase, which is returned in the Transaction Response.
- Transaction Reversal Result — indicates the result of order reversal for each authorization or purchase that occurred due to risk assessment.

Note: This feature is available on both 2-Party and 3-Party transactions.

Transaction Request Input Fields

Risk Management Input Fields			
Include the following data in addition to the required fields for a basic 2-Party or 3-Party transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_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

Bank Account Type Field

The Bank Account Type card field is applicable to card types such as Maestro. The Bank Account Type functionality allows the merchant to enter the type of account, Savings or Cheque, to be stored on the Payment Server for that transaction. Bank Account Type is passed with the Transaction Request and stored on the Payment Server.

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

Transaction Request Input Fields

Bank Account Type 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_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.			
Optional	Alphanumeric	3	SAV

Transaction Response Output Fields

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

ANZ Bank Extended OrderInfo Field

This is an extended OrderInfo field for the ANZ bank only. Some ANZ merchants require extra customer data for their records.

It is for display purposes only (in Merchant Administration) and is not to be passed in any messages to the acquirer. Merchant Administration users are able to view this extended OrderInfo data in the Orders History Detail Page.

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

Transaction Request Input Fields

ANZ Bank Extended OrderInfo 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_ANZExtendedOrderInfo			
This is an extended OrderInfo field for the ANZ bank **only**. If the extended data is not 108 bytes then it must be padded to 108 bytes using for example, a space character (ASCII Dec 32), which will not be visible in the display.			
Optional	Alphanumeric	0,108	Extra information about this transaction that will be displayed in Merchant Administration.

Transaction Response Output Fields

ANZ Bank Extended OrderInfo 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_ANZExtendedOrderInfo			
This is the vpc_ANZExtendedOrderInfo input returned.			
Input	Alphanumeric	0,108	This is some extra information about this transaction that will be displayed in merchant Administration.

CashAdvance

Adding the CashAdvance field to a normal card present purchase mode transaction causes a cash advance transaction of the specified amount to be performed. It is only valid to submit the CashAdvance Transaction Request field when the Merchant Transaction Source field (vpc_TxSource) has a value of CARDPRESENT.

Transaction Request Input Fields

AMA Cash Advance 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_CashAdvance			
Adding this field to a card present purchase mode transaction, causes the transaction to be submitted as a cash advance transaction to the value specified in the Amount field. Valid values are: ▪ **Y, Yes, True, 1** – all of the above indicate that a purchase mode transaction is to be put through as a cash advance. ▪ **Any other value** – Ignore this field. The purchase mode transaction is submitted as a normal purchase transaction.			
Required	Alphanumeric	1,4	Yes

Transaction Response Output Fields

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

Note: If using this field you must also use the Card Present Fields as well as the ***required DO Fields*** on page 17.

Verification Only

Verification Only transactions are submitted to the acquirer (if supported) as account status inquiries. If you provide a Card Security Code (CSC) and/or Billing Address details, they will be included in the request submitted to the acquirer and you may receive a CSC/Address Verification Service (AVS) validation and/or response code.

Note: Verification Only transactions are not supported for Maestro cards.

To submit a Verification Only transaction, you must be configured with the *May Use Verification Only* privilege by your Payments Services Provider.

Transaction Request

Verification Only Request Fields			
Include the following fields in the transaction request when submitting a 2-Party VPC Verification Only 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 transaction type. This must be equal to a ‘doRequest’ for a Verification Only 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 '**VERIFICATION**' for this type of transaction.			
Required	Alphanumeric	1,20	VERIFICATION

vpc_RequestCommand			
This field is associated when the **vpc_Command** field equals '**doRequest**'. The value must be equal to '**doVerificationOnly**' for this type of transaction.			
Required	Alphanumeric	1,20	doVerificationOnly
vpc_Merchant			
The unique Merchant ID assigned to you by your Payments Service Provider.			
Required	Alphanumeric	1,16	TESTMERCHANT01
vpc_AccessCode			
The access code is provided when you register with your Payments Service Provider.			
Required	Alphanumeric	8	6AQ89F3

vpc_SessionId			
An identifier for the Order. You may use this identifier to search for the order. This will be returned in the vpc_OrderInfo field in the transaction response. **Note: If the "Enforce Unique Order Reference" privilege is enabled for your profile by your Payment Provider, this value must be unique across all your orders.**			
Required	Alphanumeric	0,34	ORDER958743
vpc_MerchTxnRef			
An optional identifier for this transaction. You may use this identifier to retrieve the transaction result in Query DR and you may also use it to search for the transaction. **Note**: If the "Enforce Unique Merchant Transaction Reference" privilege is enabled for your profile by your Payment Provider, this value must be unique across all your transactions.			
Optional	Alphanumeric	1,40	ORDER958743-1
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. Note: This field is mandatory if you are configured with more than one currency.			
Conditional	Alpha	3	USD
vpc_CardNum			
The number of the card used for the transaction. The card number 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 (American Express) or CVV2, which is printed, not embossed on the card. It is used to compare it with the records held in the card issuer's database.			
Optional	Numeric	3,4	985

vpc_AVS_Street01			
The street name and number, or the Post Office Box details, of the address — may be used for Address Verification check by the card issuing bank.			
Optional	Alphanumeric	1,128	1136 John Street

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

vpc_AVS_StateProv

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

vpc_AVS_PostCode			
The Postal/Zip code of the address — may be used in the Address Verification check by the card issuing bank.			
Optional	Alphanumeric	4,9	98111

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

Transaction Response

Verification Only Output Fields			
The following fields are returned in the Transaction Response for a Verification Only transaction.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data
vpc_Version			
The value of the vpc_Version input field as provided in the request.			
Input	Alphanumeric	1,8	1

vpc_Command			
The value of the vpc_Command input field as provided in the request.			
Input	Alphanumeric	1,16	doRequest

vpc_RequestType			
The value of the vpc_RequestType input field as provided in the request.			
Input	Alphanumeric	1,20	VERIFICATION

vpc_RequestCommand			
The value of the vpc_RequestCommand input field as provided in the request.			
Input	Alphanumeric	1,20	doVerificationOnly

vpc_Merchant			
The value of the vpc_Merchant input field as provided in the request.			
Input	Alphanumeric	1,16	TESTMERCHANT01
vpc_OrderInfo			

The value of the vpc_SessionId input field as provided in the request.			
Input	Alphanumeric	1,34	ORDER958743

vpc_MerchTxnRef			
The value of the vpc_MerchTxnRef input field as provided in the request.			
Input	Alphanumeric	0,40	ORDER958743-1

vpc_ShopTransactionNo			
A unique order number generated by the Payment Server for the transaction. All subsequent transactions you perform on this transaction will be assigned the same order number.			
Output	Numeric	1,19	10712
vpc_TransactionNo			
A unique transaction ID 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 to identify the target transaction when performing subsequent transactions such as refund, capture and void. This field is not returned for transactions that result in an error condition.			
Output	Numeric	1,19	96841
vpc_Currency			
The value of the vpc_Currency input field as provided in the request.			
Input	Alpha	3	USD
vpc_Card			
Identifies the card type used for the transaction. For example, MC for Mastercard. For a full list of card types, see ***Card Type Codes*** in page 137.			
Output	Alpha	0,2	MC
vpc_AVS_Street01			
The value of the vpc_AVS_Street01 input field as provided in the request.			
Input	Alphanumeric	0,20	1136 John Street

vpc_AVS_City			
The value of the vpc_AVS_City input field as provided in the request.			
Input	Alphanumeric	0,20	Seattle

vpc_AVS_StateProv			
The value of the vpc_AVS_StateProv input field as provided in the request.			
Input	Alphanumeric	0,5	WA

vpc_AVS_PostCode			
The value of the vpc_AVS_PostCode input field as provided in the request.			
Input	Alphanumeric	0,9	98111

vpc_AVS_Country			
The value of the vpc_AVS_Country input field as provided in the request.			
Input	Alpha	0,3	USA
vpc_TxnResponseCode			

A response code that is generated by the gateway 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).			
Output	Alphanumeric	1	0
vpc_Message			
Message indicating what sort of errors the transaction encountered.			
Output	Alphanumeric	1,255	Merchant [TESTCORE23] does not exist.
vpc_AcqResponseCode			
The response code indicating the status of the transaction, as returned by the acquirer.			
Output	Alphanumeric	2,3	00
vpc_CSCResultCode			
Card Security Code (CSC) validation response code as determined by the gateway based on the code returned by the acquirer. 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 '**Unsupported**'.			
Output	Alpha	1,11	M
vpc_AcqCSCRespCode			
Card Security Code validation response code, as returned by the acquirer.			
Output	Alpha	1,11	M
vpc_AVSResultCode			
The result code generated by the gateway to indicate the AVS level that was used to match the data held by the cardholder's issuing bank. **Note:** Returned as '**Unsupported**' if the acquirer does not support AVS.			
Output	Alpha	1,11	Y
vpc_AcqAVSRespCode			
Address Verification Service (AVS) response code, as returned by the acquirer.			
Output	Alpha	1,11	Y
vpc_ReceiptNo			
A unique identifier 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_BatchNo			
Acquirer batch ID. Always set to 0. Can be ignored.			
Output	Numeric	0,8	0
vpc_RiskOverallResult			

The overall result of risk assessment for this transaction. Only returned if you are enabled for risk. Values: ACC (Accepted) — indicates that the order has been accepted. REJ (Rejected) — indicates that the order has been rejected. REV (Review Required) — indicates that the order has been flagged for review. NCK (Not Checked) — indicates that the order has been processed using the 'Bypass Risk Management' flag. It also implies a condition where neither MSO nor merchant risk rules are configured in the system. SRJ (System Rejected) — indicates that the order has been rejected at the system (MSO) level.			
Output	Alphanumeric	3	0
vpc_Locale			
The merchant's locale. Can be ignored.			
Output	Alphanumeric	2,5	en_US

Credential on File Fields

You can perform cardholder-initiated and merchant-initiated transactions using Credential on File.

Credential on File, also known as stored credentials, are account details that you collect from your cardholders, store them, and either you (merchant-initiated) or your cardholders (cardholder-initiated) use the stored account details for subsequent payments.

If you are using *Card Scheme Tokens* you can choose to store or not store them. The gateway supports flagging of transactions as Credential on File for credentials stored outside of the gateway. You can indicate if the credentials are stored, not stored, or you intend to store them using the Transaction Request Input Fields outlined below.

Cardholder-initiated Transactions

A cardholder-initiated transaction is a payment that is initiated *with* the active participation of the cardholder. It may be performed with or without using stored credentials.

Merchant-initiated Transactions

A merchant-initiated transaction is one that is performed *without* the active participation of the payer. It may be performed as a follow-up to a cardholder-initiated transaction or to execute a pre-agreed standing instruction from the cardholder for the provision of goods or services. For example, a subsequent recurring payment for a magazine subscription, auto top-up for prepaid accounts, etc.

Identifying merchant-initiated transactions can provide transaction transparency, resulting in higher authorization rates and improved cardholder experience. Only standing instructions where you have an agreement with the cardholder to debit their account (for example, installment payments, recurring payments, or unscheduled payments) are currently supported.

Note: Applies to 2-party transactions.

Transaction Request Input Fields

Credential On File 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_CardStoredOnFile			
This field only applies if you collect and store card details from your cardholder and use the stored value for subsequent payments. If the card details are not stored or you do not intend to store them, you need not provide this field. Valid values for this field are: ▪ STORED: Use this value if the card details provided have been stored previously. ▪ TO_BE_STORED: Use this value if this is the first transaction using the card and you intend to store the card details only if the transaction is successful. **Notes**: If you use card scheme tokenization services like MDES (Mastercard Digital Enablement Service) and store the tokens provided, you have to provide the value STORED, and if you pass the token without storing them, you are not required to provide this field. It's highly recommended that you flag merchant-initiated transactions correctly using this field for better approval rates.			
Optional	Alphanumeric	6,12	STORED
vpc_AgreementId			
This is a unique value generated by the merchant to identify a payment agreement with the cardholder. When you collect payment credentials from your cardholders and store them for later use, you must provide an agreement ID when you use the stored credentials for the following merchant-initiated transactions: ▪ Recurring payments (**vpc_TxSourceSubType**=**RECURRING**): You have an agreement with the cardholder that authorizes you to automatically debit their account at agreed intervals for fixed or variable amounts. For example, gym membership, phone bills, or magazine subscriptions. ▪ Installment payments (**vpc_TxSourceSubType**=**INSTALLMENT**): You have an agreement with the cardholder that authorizes you to process multiple payments over an agreed period of time for a single purchase. For example, the payer purchases an item for $1000 and pays for it in four monthly installments. ▪ Unscheduled payments (**vpc_TxSourceSubType**=**SINGLE**): You have an agreement with the cardholder that authorizes you to process future payments when required. For example, the cardholder authorizes you to process an account top-up transaction for a transit card when the account balance drops below a certain threshold.			
Optional	Alphanumeric	1,100	ABC_COF_AG_ID_001
vpc_TxSource			

The source of the transaction. You must set this to “MERCHANT” for a merchant-initiated transaction. For example, a recurring payment, installment payment, or account top-up. This is required to be set only if the merchant's default transaction source has not been configured to MERCHANT.			
Optional	Alphanumeric	11	MERCHANT
vpc_TxAcquirerTraceId			
The unique identifier that you can provide in a Purchase transaction, which allows the issuer to link related transactions, for example, merchant-initiated transactions. It is only applicable if you want to link transactions across multiple payment gateways. To find its usage, look up 'trace identifier’ or 'transaction identifier’ in the Mastercard and Visa documentation respectively. If you provide the Trace ID in the request, the Payment Server will use this value in preference to the value stored against the Agreement ID.			
Conditional	Alphanumeric	1,15	123458908123342

Transaction Response

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

Card Scheme Tokens

Card scheme tokenization services, for example, Mastercard Digital Enablement Service (MDES), enable you to store cardholder’s card details in exchange for a token. Card scheme tokens provide better security for payment information using dynamic cryptograms. They also provide an enhanced user experience, keep card information up to date, and can potentially deliver higher approval rates.

You can obtain a card scheme token by integrating directly to MDES and use the token credentials to process a payment via the Payment Server. Currently the gateway only supports processing card scheme tokens obtained from MDES.

Note: Applies to 2-Party transactions.

Transaction Request

Request Fields			
In addition to the standard fields, include the following fields in the transaction request when submitting a card scheme token in a 2-Party transaction.			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_CardNum			
The card scheme token obtained from MDES.			
Required	Numeric	15,19	5480981500100002

vpc_CardExp			
The expiry date of the card scheme token.			
Required	Numeric	4	1305
vpc_TokenPaymentIndicator			
Set this value to 'C' for a payment using a card scheme token. When you provide this field, you must also provide the cryptogram in the vpc_TokenPaymentCryptogram field. You must also set this field to 'C' for recurring transactions and partial shipments where the cryptogram only needs to be provided on the first transaction in the series.			
Optional	String	1	C
vpc_TokenPaymentCryptogram			
The cryptogram value provided by MDES. Supply this value in all requests where vpc_TokenPaymentIndicator is set to 'C' and the transaction type is not recurring or partial shipment.			
Optional	String	1, 128	12233445566778899001122334455667788991
vpc_TxShipmentType			
Set this value to 'PS' in the initial transaction for partial shipment transactions. This applies when there is an agreement to supply some goods or services, and you fulfill that agreement in multiple shipments and require payment for each shipment.			
Optional	String	2	PS
vpc_TerminalLocation			
Specifies the location of the terminal in relation to the premises of the card acceptor. Valid values are: P - A terminal under the merchant's control on the merchant's premises was used. O - A terminal under the merchant's control but not on the merchant's premises was used. D - A terminal under the payer's control on the merchant's premises was used. For example, a mobile device or personal computer. M - A terminal under the payer's control and off the merchant's premises was used. For example, a mobile device or personal computer.			
Optional	Alphanumeric	1	P
vpc_TxSource			
Set this to INTERNET or MERCHANT for a payment using a card scheme token. This is required to be set only if the merchant's default transaction source has not been configured to INTERNET/MERCHANT. Set this field to INTERNET or MERCHANT for transaction frequencies (**vpc_TxSourceSubType)** SINGLE or RECURRING/INSTALLMENT respectively.			
Optional	Alphanumeric	11	INTERNET

Transaction Response

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

Payment Authentication

The Payment Server supports payment authentication using 3-Domain Secure™ (3-D Secure or 3DS), an authentication protocol designed to reduce fraud and provide additional security to e-commerce transactions. It allows the merchant to authenticate the payer at their card issuer before submitting an Authorization or Purchase transaction.

Key Benefits

3DS offers the following benefits to the merchant:

- Fraud protection as the payer is authenticated at their card issuer.
- Liability shift — payments where 3DS is performed shift the liability to the issuer. This means if a payer disputes the payment and claims a chargeback, the liability for fraudulent chargebacks shifts from the merchant to the issuer.
- Enhanced security on payments as the payer is assessed for risk by the issuer's Access Control Server (ACS)

3DS Authentication Versions

The Payment Server supports the following versions of 3DS authentication:

- 3DS, is the original version that requires cardholders to authenticate at their issuer's Access Control Server (ACS) by responding to an authentication challenge, for example, by entering a one-time password (OTP). This authentication version is also known as **3DS1** in the Payment Server.

  Supported authentication schemes for 3DS1 include Mastercard SecureCode™, Verified by Visa™, American Express SafeKey™, J/Secure™, and Diners Club ProtectBuy™.

- EMV 3DS, is the new version designed by EMVCo and adopted by most card schemes. It is an intelligent solution that provides enhanced security in online purchases while providing frictionless checkouts to cardholders where applicable. For example, the issuer may bypass the authentication challenge if the payment is considered low risk.

  The ACS determines the risk using information provided by the merchant, browser fingerprinting, and/or previous interactions with the payer. The ACS subjects the cardholder to a challenge (for example, entering a PIN) only where additional verification is required to authenticate the cardholder. This authentication type is also known as **3DS2** in the Payment Server.

  Supported authentication schemes for 3DS2 include Mastercard SecureCode™, Verified by Visa™, and American Express SafeKey™.

Prerequisites

Before you build your integration to the Payment Server for 3DS, ensure the prerequisites are met.

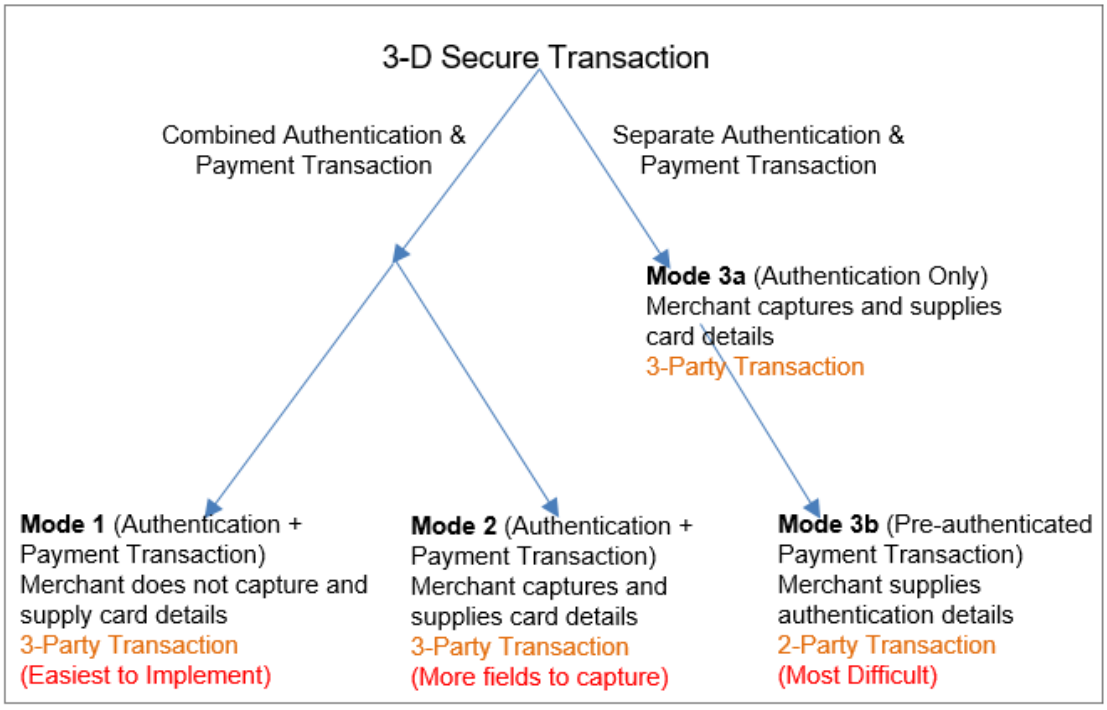
- The merchant profile on the Payment Server must be enabled for the 3DS authentication scheme and the authentication version, 3DS1 and/or 3DS2.
    - For Mastercard, Visa and American Express, the merchant can be enabled and configured for 3DS1 or 3DS2 or both.
    - For JCB and Diners, the merchant can be enabled and configured for 3DS1 only.

Note: If the merchant is enabled and configured for both 3DS versions, the Payment Server always attempts 3DS2 first, and will attempt 3DS1 (if supported by the issuer and card) only when 3DS2 is not available for the card. If neither are available, authentication will not be performed.

- Where the merchant wants the Payment Server to perform authentication, they must be enabled for 3-Party transactions.
- Where the merchant performs authentication outside of the Payment Server but wants to submit the authentication details on the transaction request, they must be enabled for the MOTO privilege.

Payment Server Integration Modes for 3DS Authentication

The following diagram shows the different modes the merchant can integrate to the Payment Server to perform 3DS authentication.



The available integration modes for 3DS authentication are:

1 ***Mode 1 - Combined 3-Party Authentication & Payment Transaction (Payment Server collects card details)***: The merchant uses the Payment Server to perform both authentication and payment.
The Payment Server collects the card details from the cardholder and performs the authentication. The Payment Server uses the authentication details when performing 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 both authentication and payment.

The merchant's application collects the cardholder's card details and sends them to the Payment Server when redirecting the cardholder. The Payment Server performs the authentication and uses the authentication details when performing the payment transaction.

3 ***Mode 3a - 3-Party Authentication Only (Merchant collects card details)*****:** The merchant's application collects the cardholder's card details and the merchant uses the Payment Server to perform the authentication.
The merchant subsequently submits a 2-Party payment request with the authentication details to the Payment Server for processing. This gives the merchant control as to when and if a payment transaction should proceed based on the result of the authentication.

4 ***Mode 3b - 2-Party Pre-Authenticated Payment Transaction (Merchant supplies authentication details)****:* The merchant performs the authentication using **Mode 3a** or an external authentication provider. The merchant subsequently submits the authentication details on a 2-Party payment request to the Payment Server.

Advantages and Disadvantages of the Integration Modes

Mode	Advantages	Disadvantages
Mode 1 3-Party Authentication and Payment transaction (Payment Server collects card details)	▪ Simple to implement. ▪ The Payment Server collects the card details from the cardholder on behalf of the merchant, which provides the highest level of security for the card details.	▪ The merchant is not able to use their own branding throughout the checkout process, as the Payment Server 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 be submitted for processing unless rejected by the 3-D Secure Risk Rules. See *Risk Management Fields*.
Mode 2 3-Party Authentication and Payment transaction (Merchant collects card details)	▪ Suits a merchant that normally collects the card details. ▪ Branding of the payment pages on the website remains consistent throughout the checkout process (except for the screen where the cardholder interacts with the issuer's ACS).	▪ If the cardholder is not enrolled in 3-D Secure the authentication will not take place and the transaction will be submitted for processing unless rejected by the 3-D Secure Risk Rules. See *Risk Management Fields*. ▪ Requires the merchant to collect card details, meaning the merchant must be PCI DSS compliant.

Mode	Advantages	Disadvantages
Mode 3a 3-Party Authentication Only (Merchant collects card details)	▪ Suits a merchant that normally collects the card details. ▪ Branding of the payment pages on the website remains consistent throughout the checkout process (except for the screen where the cardholder interacts with the issuer's ACS).	▪ Requires handling of two separate steps, the authentication and the payment, which can be more difficult for a merchant to implement. ▪ Requires the merchant to collect card details, meaning the merchant must be PCI DSS compliant.
Mode 3b 2-Party Pre-Authenticated transaction (Merchant supplies authentication details)	▪ Gives the merchant control over whether to process the transaction based on the authentication result. If the cardholder is not enrolled in 3DS or did not successfully authenticate, then the merchant's application can stop the transaction processing providing them control over how to handle the risk. ▪ Branding remains consistent throughout the checkout process (except for the one screen where the cardholder interacts with the issuer's ACS).	▪ Requires handling of two separate steps, the authentication and the payment, which can be more difficult for a merchant to implement. ▪ Requires the merchant to collect card details, meaning the merchant must be PCI DSS compliant.

Information Flow for 3DS Authentication

This section describes the information flow for a successful authentication where the Payment Server collects the card details (Mode1) and performs 3DS authentication.

Note: If the merchant is configured for both 3DS1 and 3DS2, they cannot choose the 3DS version that will be performed for a transaction request. Where 3DS2 is supported, the Payment Server always attempts 3DS2 first. Only where 3DS2 is not available for a card, the Payment Server will attempt 3DS1.

3DS1 Authentication Information Flow

The information flow for a successful authentication where the merchant is enabled for 3DS1 only, and the cardholder is enrolled for 3DS1 is as follows:

1 A cardholder browses the merchant's shop site, selects one or more products, proceeds to the checkout.

2 The cardholder confirms that they want to proceed with the payment and the merchant's application redirects the cardholder's browser to the Payment Server.

3 The Payment Server prompts the cardholder to enter the card details, and the cardholder selects to pay with a credit or debit card that supports 3DS1.

4 The Payment Server initiates the authentication and redirects the cardholder's browser to the issuer's ACS. The cardholder is prompted to respond to an authentication challenge.

5 The issuer returns the cardholder's browser to the Payment Sever and the Payment Server retrieves the authentication result from the issuer's ACS. The Payment Server processes the payment with the authentication details and redirects the cardholder back to the merchant's site.

If the payer did not authenticate successfully or is not enrolled in 3DS1, the Payment Server will proceed with processing the payment unless the transaction is blocked by 3-D Secure Risk Rules. See *Risk Management Fields*.

3DS2 Authentication Information Flow

The information flow for a successful authentication where the merchant is enabled for 3DS2 (optionally 3DS1) and the cardholder is enrolled for 3DS2 is as follows:

1 A cardholder browses the merchant's shop site, selects one or more products, proceeds to the checkout.

2 The cardholder confirms that they want to proceed with the payment and the merchant's application redirects the cardholder's browser to the Payment Server.

3 The Payment Server prompts the cardholder to enter the card details, and the cardholder selects to pay with a credit or debit card that supports 3DS2.

4 The Payment Server initiates the authentication, and the issuer determines the authentication flow based on the risk associated with the payment. The issuer may offer either of the following flows:

- Frictionless Flow: No authentication challenge is presented. The Payment Server performs the payment and redirects the cardholder back to the merchant's site.
- Challenge Flow: If the issuer requires the cardholder to respond to a challenge, the Payment Server redirects the cardholder's browser to the issuer's ACS. The cardholder is prompted to respond to an authentication challenge. The issuer returns the cardholder's browser to the Payment Server.

The Payment Server retrieves the authentication result from the issuer's ACS, processes the payment with the authentication details and redirects the cardholder back to the merchant's site.

Note: If 3DS2 is not available, the Payment Server will attempt 3DS1 (if it's available), where the cardholder will be presented with an authentication challenge, as described in *3DS1 Authentication Information Flow*.

If the payer did not authenticate successfully or is not enrolled in 3DS1 or 3DS2, the Payment Server will proceed with processing the payment unless the transaction is blocked by 3-D Secure Risk Rules. See *Risk Management Fields*.

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

In this mode, the merchant uses the Payment Server to perform both authentication and payment. The Payment Server collects the cardholder's card details.

Mode 1 Transaction Request Input Fields – 3DS1 Authentication

If the merchant is configured for 3DS1 and wants to perform 3DS1 for this transaction, they do not need to provide any additional input fields in the Transaction Request. They must simply provide the standard fields for a 3-Party transaction. See *Input Fields for Basic 3-Party Transactions*.

See also ***3DS1 Authentication Information Flow***.

Mode 1 Transaction Response Output Fields – 3DS1 Authentication

The following fields are only returned in the Transaction Response if the transaction includes authentication details. The merchant should store these details as a record of the authentication for the transaction to resolve any chargeback disputes.

The response code returned in the field **vpc_VerStatus** indicates whether the authentication was successful or not. For a list of values for this field, please see ***3-D Secure Status Codes***.

Mode 1 3DS1 Output Fields

In addition to the standard output fields (see page 24), the following fields are also returned in the Transaction Response for 3-Party transactions where authentication was initiated by the Payment Server.

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 (ECI) returned by the Access Control Server (ACS). It indicates the level of security and authentication of the transaction. Possible values depend on the card scheme. For example, if the cardholder was successfully authenticated by the issuer, the value is: ▪ 02 for Mastercard SecureCode. ▪ 05 for Verified by Visa and American Express SafeKey. If the cardholder failed authentication, the value is: ▪ 00 for MasterCard SecureCode. ▪ 07 for Verified by Visa and American Express SafeKey.			
Output	Numeric	2	05

vpc_3DSXID			
A unique transaction identifier that is generated by the Payment Server (on behalf of the merchant) to identify the 3DS transaction. This 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 based on the information provided by the scheme's Directory Server (DS). This is the value of the VERes.enrolled field returned by the DS. 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 by the Payment Server, i.e., an authentication request was submitted to the issuer's ACS and a PARes was received by the Payment Server. 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 issuer as a token for the merchant to prove that the cardholder authentication was performed. This is a base64 encoded value.			
Output	Alphanumeric	28	gIGCg4SFhoeIiYqLjI2Oj5CRkpM=

vpc_VerType			
This field will always be set to '3DS' indicating that one of the 3DS schemes was used.			
Output	Alphanumeric	0,3	3DS

vpc_VerSecurityLevel			
The Electronic Commerce Indicator (ECI) value as submitted by the Payment Server to the acquirer. Indicates the level of security and authentication of the transaction. Depending on the acquirer and the result of the authentication, this value may be different from the ECI value returned from the Access Control Server (ACS).			
Output	Numeric	0,2	06

vpc_VerStatus			
The status codes used by the Payment Server to indicate the result of the payment authentication. ***3-D Secure Status Codes*** on page 135.			
Output	Alphanumeric	1	N

vpc_3DS2dsTransactionId			
A unique identifier for the authentication assigned by the scheme's Directory Server (DS). **Note**: This field is only returned if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Conditional	Alphanumeric	1,50	211566f4-05af-48d3-967a-d68be1956d6b
vpc_AuthenticationVersion			
The 3DS version used for cardholder authentication. **Note**: This field is only returned if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Input	Numeric	1	2

Mode 1 Transaction Request Input Fields – 3DS2 Authentication

If the merchant is already supporting 3DS1 and wants to upgrade to 3DS2, they must provide the following fields in the Transaction Request in addition to the standard fields for a 3-Party transaction. See *Input Fields for Basic 3-Party Transactions*. If the merchant does not provide these fields, the Payment Server will attempt 3DS1.

See also ***3DS2 Authentication Information Flow.***

3DS2 Authentication Request Fields			
Include the following data in addition to the required fields for a basic 3-Party transaction (see *Input Fields for Basic 3-Party Transactions)*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_3ds2DataVersion			
This field must be set to 55. If not specified, the Payment Server defaults the value to 55.			
Optional	Numeric	0,2	55
vpc_3ds2AuthenticatePayer			

Additional data required by the authentication schemes to support 3DS2 cardholder authentication.

3DS2 requires a rich data set to allow the merchant to provide the best authentication experience (frictionless flow) to their cardholder. To provide this data, the merchant must populate this field with any of the data elements specified in the Web Services API **AUTHENTICATE_PAYER** operation (v55).
For example, to add the customer's phone number and mobile to the 3DS2 data, the merchant must provide the following value:

```
{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}
```

Note: It is recommended that the merchant supplies as much of this data as possible, as this increases the likelihood that the ACS will offer frictionless authentication, which greatly improves the cardholder experience resulting in a more streamlined checkout.

The merchant must use this field to provide data elements for which an equivalent VPC request field (with the vpc_ prefix) does not exist. Where a VPC request field exists, the merchant must provide the data in this existing VPC request field. For example, the merchant must provide the card expiry month and year in the field *vpc_CardExp* rather than via the Web Services API field *sourceOfFunds.provided.card.expiry* provided in the field vpc_3dsAuthenticatePayer.

If the merchant provides both the VPC request field and the corresponding Web Services API field in the field vpc_3dsAuthenticatePayer then the Payment Server ignores the Web Services API field and sources data from the VPC request field.

If the merchant provides any invalid data elements in the field vpc_3dsAuthenticatePayer, the Payment Server will return an error message.

For information on additional field restrictions, see *Device Details*.

Note: The merchant must ensure that the total URL redirect length is supported by web browsers. This ranges from 2000 to 4000 characters depending on the browsers they want to support on their web site.

Optional	JSON		{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}

vpc_Return3ds2Details			
An indicator of whether the Payment Server should return 3DS2 details (in the VPC response fields vpc_3DS2dsTransactionId and vpc_AuthenticationVersion) in the Transaction Response. Valid values are: ▪ Y - Yes ▪ N – No (this is the default value that will be applied, if the field is not provided)			
Conditional	Alpha	1	Y

Device Details

Device details are fields that contain information about the device used by the cardholder when making a payment.
The values for the fields in the **device** parameter group in the Web Services API **AUTHENTICATE_PAYER** operation (v55) are automatically detected by the Payment Server and populated in the Transaction Request. If the merchant provides these fields in the vpc_3ds2AuthenticatePayer field in the Transaction Request, the Payment Server rejects the request.

This includes the following Web Services API fields:

- device.browser
- device.browserDetails.3DSecureChallengeWindowSize
- device.browserDetails.acceptHeaders
- device.browserDetails.colorDepth
- device.browserDetails.javaEnabled
- device.browserDetails.language
- device.browserDetails.screenHeight
- device.browserDetails.screenWidth
- device.browserDetails.timeZone
- device.ipAddress

Browser's IP Address

The merchant can provide the IP address of the cardholder's browser in the vpc_CustomerIpAddress field. If not provided, the IP Address will be automatically detected by the Payment Server and populated in the Transaction Request.

Mode 1 Transaction Response Output Fields – 3DS2 Authentication

In addition to the *Mode 1 Transaction Response Output Fields – 3DS1 Authentication,* the following fields are retuned for 3DS2 authentication.

Mode 1 3DS2 Authentication Output Fields			
The following fields are returned in the Transaction Response for 3-Party transactions where 3DS2 authentication was performed.			
Field Name			
Field Description			
Returned Input or Output	Field Type	Min, Max or Set Field Length	Sample Data

vpc_3DS2dsTransactionId			
A unique identifier for the 3DS authentication assigned by the scheme's Directory Server (DS). **Note**: This field is returned only if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Conditional	Alphanumeric	1,50	211566f4-05af-48d3-967a-d68be1956d6b
vpc_AuthenticationVersion			
The 3DS version used for cardholder authentication. **Note**: This field is returned only if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Input	Numeric	1	2

Mode 2 - Combined 3-Party Authentication and Payment transaction (Merchant collects card details)

In this mode, the merchant's application collects the cardholder's card details and sends them to the Payment Server when redirecting the cardholder.

The merchant must be enabled for “External Pay Select” and “Card Details in Digital Order” privileges by their Payment Provider to be able to use Mode 2.

Mode 2 Transaction Request Input Fields – 3DS1 Authentication

The merchant must submit the card details collected on their website to the Payment Server using the 3-Party request fields outlined below.

Card Details in Transaction Request Fields			
Include the following data in addition to the required fields for a basic 3-Party transaction (see *Input Fields for Basic 3-Party Transactions)*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_card			
The type of card to be used for the transaction. This field is case sensitive, and must contain a valid value as defined in section **External Payment Selection (EPS)**. To identify the card types available for the merchant, they can perform a 3-Party transaction and go to the Payment Server card selection page, and run the cursor over each card logo that is displayed. The 'card' and 'gateway' values are displayed at the bottom of the browser window.			
Required	Alphanumeric	3,16	Visa
vpc_gateway			
This field must be set to **ssl** to indicate that the Payment Server should perform 3DS authentication (where applicable) and proceed with the payment. This field is case sensitive.			
Required	Alphanumeric	3,15	ssl
vpc_CardNum			
The card number to be 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. **Note**: This field is optional for Maestro cards. If you do not provide a value, the field defaults to 4912 (Dec 2049).			
Required	Numeric	4	1305

vpc_CardSecurityCode			
The Card Security Code (CSC), also known as CVV(Visa), CVC2(Mastercard) or CID/4DBC(American Express) or CVV2, which is printed, not embossed on the card. If provided, the issuer may compare the code with the records held in their database. **Note**: This field is optional for Maestro cards, even if CSC is enforced by Payment Server.			
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 only be displayed on the Verified by Visa™ page where the cardholder is requested to authenticate.			
Optional	Alphanumeric	0,125	This is a description for the transaction.

Mode 2 Transaction Response Output Fields – 3DS1 Authentication

The outputs from this transaction type are the same as *Mode 1 Transaction Response Output Fields – 3DS1 Authentication.*

Mode 2 Transaction Request Input Fields – 3DS2 Authentication

If the is merchant already supporting 3DS1 and wants to upgrade to 3DS2, the merchant must provide the following fields in addition to the 3DS1 fields, see *Mode 2 Transaction Request Input Fields – 3DS1 Authentication.* If the merchant does not provide these fields, the Payment Server will attempt 3DS1.

In addition to providing these fields, the merchant must use the 3-D Secure JavaScript API on their payment page to trigger the ACS Method call required for 3DS2. See *3-D Secure JavaScript API Integration* for details.

Card Details in Transaction Request Fields			
The merchant must provide the following data in addition to the required fields for *Mode 2 Transaction Request Input Fields – 3DS1 Authentication*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_3ds2DataVersion			
This field must be set to 55. If not specified, the Payment Server defaults the value to 55.			
Optional	Numeric	0,2	55

vpc_3ds2AuthenticatePayer

Additional data required by the authentication schemes to support 3DS2 cardholder authentication.

3DS2 requires a rich data set to allow the merchant to provide the best authentication experience (frictionless flow) to their cardholder. To provide this data, the merchant must populate this field with any of the data elements specified in the Web Services API **AUTHENTICATE_PAYER** operation (v55).
For example, to add the customer's phone number and mobile to the 3DS2 data, the merchant must provide the following value:

```
{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}
```

Note: It is recommended that the merchant supplies as much of this data as possible, as this increases the likelihood that the ACS will offer frictionless authentication, which greatly improves the cardholder experience resulting in a more streamlined checkout.

The merchant must use this field to provide data elements for which an equivalent VPC request field (with the vpc_ prefix) does not exist. Where a VPC request field exists, the merchant must provide the data in this existing VPC request field. For example, the merchant must provide the card expiry month and year in the field *vpc_CardExp* rather than via the Web Services API field *sourceOfFunds.provided.card.expiry* provided in the field vpc_3dsAuthenticatePayer.

If the merchant provides both the VPC request field and the corresponding Web Services API field in the field vpc_3dsAuthenticatePayer then the Payment Server ignores the Web Services API field and sources data from the VPC request field.

If the merchant provides any invalid data elements in the field vpc_3dsAuthenticatePayer, the Payment Server will return an error message.

For information on additional field restrictions, see *Device Details*.

Note: Ensure that your total URL redirect length is supported by web browsers. This ranges from 2000 to 4000 characters depending on the browsers you want to support on your web site.

Optional	JSON		{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}

vpc_Return3ds2Details

An indicator of whether the Payment Server should return 3DS2 details (in the VPC response fields vpc_3DS2dsTransactionId and vpc_AuthenticationVersion) in the Transaction Response. Valid values are:

- Y - Yes
- N – No (this is the default value that will be applied, if the field is not provided)

Conditional	Alpha	1	Y

Mode 2 Transaction Response Output Fields – 3DS2 Authentication

The outputs from this transaction type are the same as Mode 1 Transaction Response Output Fields – 3DS1 Authentication.

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

In this mode, the merchant performs the cardholder authentication separately to the payment. This allows the merchant to only proceed with the payment if the liability shifts to the issuer.

In Mode 1 and Mode 2, the Payment Server processes the payment if the cardholder is not enrolled in 3-D Secure. In Mode 3a, if the cardholder is not enrolled they are returned to the merchant's site where the merchant gets to decide if they want to proceed with the payment.

To subsequently perform a payment, the merchant must use the authentication details provided in the authentication response as additional inputs to a 2-Party transaction.

The merchant must be enabled for "External Pay Select" and "Card Details in Digital Order" privileges by their Payment Provider to be able to use Mode 3a.

Mode 3a Authentication Only Input Fields – 3DS1 Authentication

The merchant must provide the following fields in addition to the standard fields for a basic 3-Party transaction ((see *Input Fields for Basic 3-Party Transactions)*. **No payment transaction is submitted to the Payment Server** as part of the 3-Party interaction.

Authentication Only Fields			
Include the following data in addition to the required fields for a basic 3-Party transaction (see *Input Fields for Basic 3-Party Transactions)*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_card			
The type of card to be used for the transaction. This field is case sensitive, and must contain a valid value as defined in section **External Payment Selection (EPS)**. To identify the card types available for the merchant, they can perform a 3-Party transaction and go to the Payment Server card selection page, and run the cursor over each card logo that is displayed. The 'card' and 'gateway' values are displayed at the bottom of the browser window.			
Required	Required	Required	Required
vpc_gateway			
The merchant must set this field to **threeDSecure** to indicate that the Payment Server must only perform the 3DS authentication, but not proceed with the payment. The field is case sensitive.			
Required	Alphanumeric	3,15	threeDSecure
vpc_CardNum			
The card number to be 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. **Note**: This field is optional for Maestro cards. If you do not provide a value, the field defaults to 4912 (Dec 2049).			
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 only be displayed on the Verified by Visa™ page where the cardholder is requested to authenticate.			
Optional	Alphanumeric	0,125	This is some description about the Verified by Visa™ transaction.

Mode 3a Authentication Only Output Fields – 3DS1 Authentication

The outputs from this transaction type are the same as *Mode 1 Transaction Response Output Fields – 3DS1 Authentication*.

Mode 3a Authentication Only Input Fields – 3DS2 Authentication

If the merchant is already supporting 3DS1 and wants to upgrade to 3DS2, the merchant must provide the following fields in addition to the 3DS1 fields, see *Mode 3a Authentication Only Input Fields – 3DS1 Authentication*. If the merchant does not provide these fields, the Payment Server will attempt 3DS1.

In addition to providing these fields, the merchant must use the 3-D Secure JavaScript API on their payment page to trigger the ACS Method call required for 3DS2. See *3-D Secure JavaScript API Integration* for details.

Authentication Only Fields			
Include the following data in addition to the required fields for *Mode 3a Authentication Only Input Fields – 3DS1 Authentication*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_3ds2DataVersion			
This field must be set to 55. If not specified, the Payment Server defaults the value to 55.			
Optional	Numeric	0,2	55

vpc_3ds2AuthenticatePayer

Additional data required by the authentication schemes to support 3DS2 cardholder authentication.

3DS2 requires a rich data set to allow the merchant to provide the best authentication experience (frictionless flow) to their cardholder. To provide this data, the merchant must populate this field with any of the data elements specified in the Web Services API **AUTHENTICATE_PAYER** operation (v55).
For example, to add the customer's phone number and mobile to the 3DS2 data, the merchant must provide the following value:

```
{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}
```

Note: It is recommended that the merchant supplies as much of this data as possible, as this increases the likelihood that the ACS will offer frictionless authentication, which greatly improves the cardholder experience resulting in a more streamlined checkout.

The merchant must use this field to provide data elements for which an equivalent VPC request field (with the vpc_ prefix) does not exist. Where a VPC request field exists, the merchant must provide the data in this existing VPC request field. For example, the merchant must provide the card expiry month and year in the field *vpc_CardExp* rather than via the Web Services API field *sourceOfFunds.provided.card.expiry* provided in the field vpc_3dsAuthenticatePayer.

If the merchant provides both the VPC request field and the corresponding Web Services API field in the field vpc_3dsAuthenticatePayer then the Payment Server ignores the Web Services API field and sources data from the VPC request field.

If the merchant provides any invalid data elements in the field vpc_3dsAuthenticatePayer, the Payment Server will return an error message.
For information on additional field restrictions, see *Device Details*.

Note: Ensure that your total URL redirect length is supported by web browsers. This ranges from 2000 to 4000 characters depending on the browsers you want to support on your web site.

Optional	JSON		{"customer":{"phone":"+61733691372", "mobilePhone":"+6143808251672"}}

vpc_Return3ds2Details

An indicator of whether the Payment Server should return 3DS2 details (in the VPC response fields vpc_3DS2dsTransactionId and vpc_AuthenticationVersion) in the Transaction Response. Valid values are:

- Y - Yes
- N – No (this is the default value that will be applied, if the field is not provided)

Conditional	Alpha	1	Y

Mode 3a Authentication Only Output Fields – 3DS2 Authentication

The outputs from this transaction type are the same as *Mode 1 Transaction Response Output Fields – 3DS1 Authentication*.

Mode 3b - 2-Party Pre-Authenticated Payment Transaction (Merchant supplies authentication details)

Where the cardholder has been authenticated using Mode 3a, and where the merchant wants to use the authentication details to perform a payment, the merchant must submit a 2-Party request with additional fields containing the authentication details.

Mode 3b Transaction Request Input Fields – 3DS1 Authentication

Where the cardholder was authenticated using 3DS1, the following fields must be provided in addition to a standard fields required for a 2-Party transaction, see *Input Fields for Basic 2-Party Transactions*.

3DS1 Authentication Payment Fields			
Include the following data in addition to the required fields for a basic 2-Party transaction (see *Input Fields for Basic 2-Party Transactions)*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data

vpc_VerType			
This field must be set to '**3DS**'.			
Required	Alphanumeric	3	3DS

vpc_VerToken			
This value is generated by the card issuer as a token to prove the cardholder authentication. This is a base64 encoded value.			
Required	Alphanumeric	28	gIGCg4SFhoeliYqLjI2Oj5CRkpM=

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

vpc_3DSECI

The 3-D Secure Electronic Commerce Indicator (ECI) returned by the Access Control Server (ACS). It indicates the level of security and authentication of the transaction.

Possible values depend on the card scheme. For example, if the cardholder was successfully authenticated by the issuer, the value is:

- 02 for Mastercard SecureCode.
- 05 for Verified by Visa and American Express SafeKey.

If the cardholder failed authentication, the value is:

- 00 for MasterCard SecureCode.
- 07 for Verified by Visa and American Express SafeKey.

Note 1: If the ECI value returned in the authentication response is '07' or '00', do NOT provide this field. For these values, the Payment Server will calculate the ECI based on the other 3DS data provided by the merchant.

Note 2: If provided, the ECI value MUST be exactly as returned in the authentication response, in a 2 digit format, i.e., the leading zero must not be removed.

Required	Alphanumeric	2	05

vpc_3DSenrolled			
This field is mandatory if the card is enrolled for 3DS, based on the information provided by the scheme's Directory Service (DS). This is the value of the VERes.enrolled field returned by the DS. It will take values (**Y** - Yes, **N** - No, **U** - Unavailable for Checking).			
Conditional	Alphanumeric	1	Y

vpc_3DSstatus			
This field must only be included if cardholder authentication was attempted by the Payment Server, i.e., an authentication request was submitted to the issuer's ACS and a PARes was received. It will take values (**Y** - Yes, **N** - No, **A** - Attempted Authentication, **U** - Unavailable for Checking).			
Conditional	Alpha	0,1	Y

vpc_AuthenticationVersion			
The 3DS version for cardholder authentication. Set this value to 1 for 3DS1.			
Conditional	Numeric	1	1

Mode 3b Transaction Response Output Fields – 3DS1 Authentication

Payment Authentication Output Fields
In addition to the standard output fields (see page 24), the following fields are returned in the Transaction Response for 2-Party transactions where authentication details were provided in the request.
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 (ECI) returned by the Access Control Server (ACS). It indicates the level of security and authentication of the transaction. Possible values depend on the card scheme. For example, if the cardholder was successfully authenticated by the issuer, the value is: ▪ 02 for Mastercard SecureCode. ▪ 05 for Verified by Visa and American Express SafeKey. If the cardholder failed authentication, the value is: ▪ 00 for MasterCard SecureCode ▪ 07 for Verified by Visa and American Express SafeKey.			
Output	Numeric	2	05

vpc_3DSXID			
A unique transaction identifier generated by the Payment Server on behalf of the merchant to identify the 3DS authentication. This 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 enrolled in 3DS, based on the information provided by the scheme's Directory Service (DS). This is the value of the VERes.enrolled field returned by the DS. It will take values (**Y** - Yes, **N** - No, **U** - Unavailable for Checking).			
Output	Alpha	1	N

vpc_3DSstatus			
This field is only included if cardholder authentication was attempted by the Payment Server, i.e., an authentication request was submitted to the issuer's ACS and a PARes was received by the Payment Server. 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 the cardholder authentication. This is a base64 encoded value.			
Output	Alphanumeric	28	gIGCg4SFhoeliYqLjI2Oj5CRkpM=

vpc_VerType			
This field will always be set to '3DS' indicating that 3DS authentication was performed.			
Output	Alphanumeric	0,3	3DS

vpc_VerStatus

The status codes used by the Payment Server to show the result of the payment authentication. ***3-D Secure Status Codes*** on page 135.			
Output	Alphanumeric	1	N

vpc_VerSecurityLevel			
The Electronic Commerce Indicator (ECI) value as submitted by the Payment Server to the acquirer. Indicates the level of security and authentication of the transaction. Depending on the acquirer and the result of the authentication, this value may be different from the ECI value returned by the Access Control Server (ACS).			
Output	Numeric	0,2	06
vpc_AuthenticationVersion			
The 3DS version used for cardholder authentication. **Note**: This field is returned only if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Input	Numeric	1	2
vpc_3DS2dsTransactionId			
A unique identifier for the 3DS authentication assigned by the scheme's Directory Server (DS). **Note**: This field is returned only if vpc_Return3ds2Details=Y was provided in the request and 3DS2 was performed.			
Conditional	Alphanumeric	1,50	211566f4-05af-48d3-967a-d68be1956d6b

Mode 3b Transaction Request Input Fields – 3DS2 Authentication

Where the cardholder was authenticated using 3DS2, provide the following fields in addition to the standard fields for a 2-Party transaction, see *Input Fields for Basic 2-Party Transactions.*

3DS2 Pre-authentication Payment Fields			
Include the following data in addition to the required fields for a basic 2-Party transaction (see *Input Fields for Basic 2-Party Transactions)*			
Field Name			
Field Description			
Required/ Optional	Field Type	Min, Max or Set Field Length	Sample Data
vpc_VerType			
This field must be set to '**3DS**'.			
Required	Alphanumeric	3	3DS
vpc_VerToken			
This value is generated by the card issuer as a token to prove the cardholder authentication. This is a base64 encoded value.			
Required	Alphanumeric	28	gIGCg4SFhoeliYqLjI2Oj5CRkpM=
vpc_3DSECI			
The 3-D Secure Electronic Commerce Indicator (ECI) returned by the Access Control Server (ACS). It indicates the level of security and authentication of the transaction. Possible values depend on the card scheme. For example, if the cardholder was successfully authenticated by the issuer, the value is: ▪ 02 for Mastercard SecureCode. ▪ 05 for Verified by Visa and American Express SafeKey. If the cardholder failed authentication, the value is: ▪ 00 for MasterCard SecureCode. ▪ 07 for Verified by Visa and American Express SafeKey. **Note 1**: If the ECI value returned in the authentication response is '07' or '00', do NOT provide this field. For these values, the Payment Server will calculate the ECI based on the other 3DS data provided by the merchant. **Note 2**: If provided, the ECI value MUST be exactly as returned in the authentication response, in a 2 digit format, i.e., the leading zero must not be removed.			
Required	Alphanumeric	2	05
vpc_AuthenticationVersion			
The 3DS version for cardholder authentication. Set this value to 2 for 3DS2.			
Conditional	Numeric	1	2
vpc_3DS2dsTransactionId			

A unique identifier for the 3DS authentication assigned by the scheme's Directory Server, and returned in the authentication response. The value for this field must be provided unaltered. **Note**: This field is returned only if vpc_Return3ds2Details=Y and 3DS2 was performed.			
Conditional	Alphanumeric	1,50	211566f4-05af-48d3-967a-d68be1956d6b

vpc_Return3ds2Details			
An indicator of whether the Payment Server should return 3DS2 details (in the VPC response fields vpc_3DS2dsTransactionId and vpc_AuthenticationVersion) in the Transaction Response. Valid values are: ▪ Y - Yes ▪ N – No (this is the default value that will be applied, if the field is not provided)			
Conditional	Alpha	1	Y

Mode 3b Transaction Response Output Fields – 3DS2 Authentication

The outputs from this transaction type are the same as *Mode 3b Transaction Response Output Fields – 3DS1 Authentication.*

3-D Secure JavaScript API Integration

With Mode 2 and Mode 3a, the merchant's application must collect the cardholder's card details and send them to the Payment Server. To be able to use 3DS2 in these modes, the merchant must implement the 3-D Secure JavaScript API to trigger the 3DS2 Method call from the cardholder's browser.

3DS2 requires the ACS to perform a Method call to gather additional data about the cardholder before performing the authentication. This will increase the likelihood of a frictionless authentication flow being available to the cardholder. The merchant must provide this additional cardholder data using the **vpc_3ds2AuthenticatePayer** request field.

Note: The Method call requires the card number, see *Mode 2 Transaction Request Input Fields – 3DS2 Authentication* or *Mode 3a Authentication Only Input Fields – 3DS2 Authentication.*

To allow the Method call to complete before 3-Party Pages attempts to authenticate the cardholder, it is recommended to execute the 3-D Secure JavaScript at the earliest opportunity in the checkout process. This will typically be when the cardholder completes entering their card number on the checkout page.

Integration Steps

1. Include the 3-D Secure JavaScript library (threeds2.bundle.js) hosted by the Payment Server in the merchant's checkout page. See *threeds2.js API Reference*.
2. Submit the **initialize( )** call to initialize the 3DS2 interaction. This must be triggered at the earliest opportunity in the checkout flow, for example, in the Document's **onload** event.
    - When you receive a success callback for the **initialize( )** call, you can submit the invokeMethod( ) call (step 2). See *Callbacks*.
    - A failure callback for the **initialize( )** call indicates failure. See *Error Codes* for the various error codes returned for the **initialize( )** call and the actions you need to perform.
3. Submit the **invokeMethod( )** call once the cardholder has entered the card number, i.e., the success callback for the **initialize( )** call has been called and the card number input field has lost focus. This call validates the card number and also invokes the Method call.
    - When you receive a success callback for the **invokeMethod( )** call, you can submit the payment for processing (once the cardholder has indicated they want to proceed with the payment by clicking the Pay button)
    - A failure callback for the **invokeMethod( )** call will result in a fallback to 3DS1 (if the merchant is enabled for 3DS1). If the merchant is not enabled for 3DS1, authentication (3DS1 or 3DS2) will not proceed. See *Error Codes* for the various error codes returned for the **invokeMethod( )** call.
4. Submit the payment once you receive the success or failure callback for the **invokeMethod( )** call, and the cardholder is redirected to the 3-Party Pages.
    - If the cardholder has clicked on 'Pay' but you have not yet received the success callback for the **invokeMethod( )** call, your application may have to wait. You may want to display an indicator to indicate that an action is in progress during this wait time.
    - In the Transaction Request for the payment, ensure the request contains the following fields to provide the cardholder with a frictionless authentication flow where possible (see *Mode 2 Transaction Request Input Fields).*
        - vpc_3ds2DataVersion=55 (if not specified, it defaults to 55)
        - vpc_3ds2AuthenticatePayer

threeds2.js API Reference

The threeds2.js JavaScript library checks if 3DS2 authentication is available for the card, and implements the 3DS2 Method interaction between the browser and ACS as part of initiating the 3DS2 authentication.

URL

https://<YOUR_VPC_URL>/psp/gateway/common/default/threeds2.bundle.js

Once imported, the script adds a global variable **threedsMethod** to the **Window** object.

Functions

initialize( )

Initializes the 3DS2 interaction.

Usage

```
initialize(data, [success], [fail])
```

Example

```
window.onload = function  () {

  var initializeData = {
      url: "https://<vpcm_host_here>/",
      parentId: "method",
      vpc_Merchant: "<your_merchant_id_here>"
    };

  function initializeSuccess() {
    console.log("initialize success");

    // Add code here to attach an onChange event handler of a card number input
field.
    // The event handler needs to call threedsMethod.invokeMethod( ).
  }

  function initializeFail(errorCode, errorMessage) {
    console.log("initialize failed: " + errorCode + " : " + errorMessage);

    // code an alternative course of action as you won't be able to proceed
    // with using threedsMethod.invokeMethod
  }

  threedsMethod.initialize(initializeData, initializeSuccess, initializeFail);
}
```

Arguments

data Object *Required*

The data configuration object describing the URL, parent ID, and the merchant ID.

data.url String *Required*

Virtual Payment Client URL. The trailing slash symbol ('/') can be provided but is not mandatory.

data.parentId String *Required*

ID of the HTML element (it is recommended to use a hidden DIV) into which the JavaScript API will inject the HTML code performing the 3DS2 Method interaction. The call will fail if an element with this ID does not exist. The leading hash symbol ('#') can be provided but is not mandatory.

data.vpc_Merchant String (1-40 characters) *Required*

The merchant's unique identifier assigned by the Payment Server.

success JavaScript function reference *Optional*

Success callback function. See *success Callback*.

fail JavaScript function reference *Optional*

Failure callback function. See *fail Callback*.

invokeMethod( )

Validates the card number entered by the cardholder and invokes the 3DS2 ACS Method interaction.

Usage

```
invokeMethod(data, [success], [fail])
```

Example

```
function invokeMethodSuccess() {
  console.log("invokeMethod success");
}

function invokeMethodFail(errorCode, errorMessage) {
  console.log("invokeMethod failed: " + errorCode + " : " + errorMessage);
}

// The code below would typically be attached to an onChange event handler
// of a card number input field, and only after the initialize success callback.

threedsMethod.invokeMethod(
  {
    vpc_Amount: <your_vpc_Amount_value_here>,
    vpc_Currency: '<your_vpc_Currency_value_here>',
    vpc_CardNum: '<the_value_from_your_card_number_input_field_here',
  },
  invokeMethodSuccess,
  invokeMethodFail,
);
```

Arguments

data Object *Required*

The data object describing the payment details: amount, currency, and card number.

data.vpc_CardNum Numeric (15-19 digits) *Required*

Card number as supplied by the cardholder. It must not contain any white space or formatting characters.

data.vpc_Amount Numeric (1-10 digits) *Optional*

Amount 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.

data.vpc_Currency String (3 characters) *Required*

Currency expressed as an ISO 4217 alphanumeric code. This field is case-sensitive and must include uppercase characters only.

success JavaScript function reference *Optional*

Success callback function. See *success Callback*.

fail JavaScript function reference *Optional*

Failure callback function. See *fail Callback*.

Callbacks

Both **initialize( )** and **invokeMethod( )** functions accept optional 'fail' and 'success' parameters, which are JavaScript function references.

success Callback

The success callback is invoked when a function wants to inform the client about success. This callback is invoked without any input parameters.

Arguments

None.

fail Callback

The fail callback is invoked when a function wants to inform the client about a failure.

Arguments

error String *Optional*

Type of error. Possible values: ERROR, INVALID_REQUEST, INVALID_CARD. See *Error Codes*.

explanation String *Optional*

Human readable description of the error. See *Explanation Values.*

Error Codes

Error Code	Operation	Description	Recommended Merchant Action
ERROR	initialize( ), invokeMethod( )	Unrecoverable error during the function invocation, for example, 3DS2 cannot be offered on 3-Party pages because the Method call has not completed, 3DS2 is not supported for the card, timeout, etc.	Continue with the payment, 3-Party pages will attempt 3DS1 (if the merchant is enabled and configured for 3DS1) If the merchant is not enabled for 3DS1, then 3-Party Pages will not

			attempt 3DS1. If the merchant requires 3DS on the transaction they may want to offer the cardholder the option to try another card.
INVALID_REQUEST	initialize( ), invokeMethod( )	Error in the merchant's profile configuration.	Check that the merchant is enabled and configured for a 3DS2 scheme.
		Validation of an input parameter has failed, for example, a mandatory parameter was not provided, **invokeMethod( )** is called before **initialize( )** function.	Check the request parameters.
INVALID_CARD	invokeMethod( )	The card number is invalid, for example, Luhn check fails, etc.	Offer the cardholder the option to re-enter the card number or try another card.

Explanation Values

The following error codes and explanation values returned by the 'fail' callback may be useful for troubleshooting purposes.

Explanation Value	Error Code	Operation	Description
data.url, data.parentId and data.vpc_Merchant must be provided	INVALID_REQUEST	initialize( )	JSON object provided as the 'data' input parameter to the initialize( ) function must have 'url', 'parentId' and 'vpc_Merchant' fields.
'parentId' field of the JSON object provided as the 'data' input parameter of the initialize( ) function must be a valid ID of an HTML that exists on the webpage.	INVALID_REQUEST	initialize( )	Unable to find HTML element with <parentId>
API must be initialized first	INVALID_REQUEST	invokeMethod( )	invokeMethod( ) is called before initialize( )
data.vpc_CardNum and data.vpc_Currency must be provided	INVALID_REQUEST	invokeMethod( )	JSON object provided as the 'data' input parameter to the invokeMethod( ) function must have 'vpc_CardNum' and

			'vpc_Currency' fields.
Card number provided (vpc_CardNum) is invalid	INVALID_CARD	invokeMethod( )	'vpc_CardNum' field of the JSON object provided as the 'data' input parameter to the invokeMethod( ) function is not a valid card number.
Unexpected server error	ERROR	initialize( ) invokeMethod( )	Error response from Payment Server
Merchant profile is not enabled for 3DS2	INVALID_REQUEST	invokeMethod( )	The merchant profile is not enabled for 3DS2
3DS2 not supported for this card	ERROR	invokeMethod( )	3DS2 is not supported for this card
Proof of work validation failed	ERROR	invokeMethod( )	The JS API failed to compute the correct key for a password provided by the VPC server. Please contact customer support.
Merchant (vpc_Merchant) does not exist	INVALID_REQUEST	initialize( )	The merchant with ID provided in vpc_Merchant field does not exist in the Payment Server.

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_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			
A unique transaction ID 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 to identify the target transaction when performing subsequent transactions such as refund, capture and void. 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_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*** on page 137. 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_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
vpc_ShopTransactionNo			
A unique order number generated by the Payment Server for the transaction. All subsequent transactions you perform on this transaction will be assigned the same order number.			
Output	Numeric	1,19	10712

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
vpc_TransNo			
Provide the value returned in the vpc_TransactionNo field for the authorisation transaction you wish to capture.			
Note: This field must be used in subsequent transactions only.			
Required	Numeric	1,19	10712

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_TransNo			
Provide the value returned in the vpc_TransactionNo field for the original authorization/purchase transaction associated with this refund.			
Note: This field must be used in subsequent transactions only.			
Required	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 you perform a Void Authorisation 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	1,17	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
vpc_TransNo			
Provide the value returned in the vpc_TransactionNo field for the authorization transaction you wish to void. The value is a unique transaction ID (generated by the Payment Server) for the authorization transaction.			
Note: This field must be used in subsequent transactions only.			
Required	Numeric	1,19	10712

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 funds from the last 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 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
vpc_TransNo			
Provide the value returned in the vpc_TransactionNo field for the original authorization transaction associated with the capture you are attempting to void. This value is the same as the value returned in the vpc_ShopTransactionNo field for the capture transaction.			
Note: This field must be used in subsequent transactions only.			
Required	Numeric	1,19	10712

Transaction Response Output Fields

2-PartyVoid Capture Output Fields			
The following additional data fields are returned in a Transaction Response for a Void 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 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	voidRefund
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_TransNo			
Provide the value returned in the vpc_TransactionNo field for the original authorization/purchase transaction associated with the refund you are attempting to void. This value is the same as the value returned in the vpc_ShopTransactionNo field for the refund transaction. **Note**: This field must be used in subsequent transactions only.			
Required	Numeric	1,19	10712

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 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
vpc_TransNo			
Provide the value returned in the vpc_TransactionNo field for the purchase transaction you wish to void. The value is a unique transaction ID (generated by the Payment Server) for the purchase transaction.			
Note: This field must be used in subsequent transactions only.			
Required	Numeric	1,19	10712

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 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 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

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

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.			
Note: This field is optional for Maestro card transactions. If you do not provide a value, the field defaults to 4912 (Dec 2049).			
Required	Numeric	4	1305

vpc_CardIssueNumber			
The issue number of the card used with cards such as Maestro and 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.			
Optional	Numeric	4	1305
vpc_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.			
Optional	Alphanumeric	3	SAV
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

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

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 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			

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. **Note**: This field is optional for Maestro card transactions. If you do not provide a value, the field defaults to 4912 (Dec 2049).			
Required	Numeric	4	1305
vpc_CardSecurityCode			
The Card Security Code (CSC), also known as CVV(Visa), CVC2(Mastercard) or CID/4DBC(American Express) 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. **Note**: This field is optional for Maestro card transactions, even if CSC is enforced.			
Optional	Numeric	3,4	985
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.			
Optional	Numeric	4	1305

vpc_CardIssueNumber			
The issue number of the card used with cards such as Maestro and Solo.			
Optional	Numeric	0,2	01

vpc_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.			
Optional	Alphanumeric	3	SAV
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

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

Transaction Response Output Fields

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

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 98 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.

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

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

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

vpc_Amount			
The value of the vpc_Amount input field returned in the Transaction Response.			
Input	Numeric	1,10	1250
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_Command			
The value of the vpc_Command input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	pay
vpc_Locale			
The value of the vpc_Locale input field returned in the Transaction Response.			
Input	Alpha	2,5	en
vpc_Merchant			
The value of the vpc_Merchant input field returned in the Transaction Response.			
Input	Alphanumeric	1,16	TESTMERCHANT01
vpc_TransactionNo			
A unique transaction ID 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 to identify the target transaction when performing subsequent transactions such as refund, capture and void. This field is not returned for transactions that result in an error condition.			
Output	Numeric	1,19	96841

CHAPTER 7

References - Virtual Payment Client

Generating a Secure Hash

Merchant integrations are required to generate a secure hash using the SHA-256 HMAC algorithm.

Creating a SHA-256 HMAC Secure Hash

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.

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.

Merchant- Supplied Parameters

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

- 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.

In summary, only parameters with user_ and vpc_ prefixes are included in the Secure Hash 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 names. Specifically, the sort order is:
- ascending order of parameter names 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. vpc_SecureHashType request field **must** be set to 'SHA256'.

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	MastercardITESTMERCHANT
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=Mastercard
ITESTMERCHANT&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:

```
ffad3a7db59cf91963ac1e53aa08b97e878c498e13fcc1de0a20b9a8e0e3eff9
```

And the resultant Request is (note the vpc_SecureHash and the vpc_SecureHashType 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=Mastercard
ITESTMERCHANT&vpc_Version=1&vpc_SecureHash=ffad3a7db59cf91963ac1e53aa08b97e878c498e
13fcc1de0a20b9a8e0e3eff9vpc_SecureHashType=SHA256
```

The Payment Server 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 the Payment Server. 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 indicates the result of attempting to perform a transaction. 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:

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
?	Response Unknown	-	-	-	-	-
0	Transaction Successful	00	00	00	00	Approved or completed successfully
		08	08	08	08	Honor with identification
		16	-	16	-	Approved, update Track #3
1	Transaction could not be processed	-	06	-	06	Error
		09	-	09	-	Request in progress
		10	10	10	10	Approved for partial amount
		11	11	11	11	Approved VIP
		12	12	12	12	Invalid transaction
		13	13	13	13	Invalid amount
		-	14	-	14	Invalid card number
		17	17	17	17	Customer cancellation
		18	18	18	18	Customer dispute
		20	20	20	20	Invalid response
		21	-	21	-	No action taken
		22	22	22	22	Suspected malfunction
		23	23	23	23	Unacceptable transaction fee
		24	24	24	24	File update not supported by receiver
		-	25	-	25	Unable to locate record on file
		26	26	26	26	Duplicate file update record, old record replaced
		27	27	27	27	File update field edit error
		28	28	28	28	File update file locked out
		29	29	29	29	File update not successful, contact acquirer
		30	30	30	30	Format error

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
		32	32	32	32	Completed partially
		35	35	35	35	Card acceptor contact acquirer
		37	37	37	37	Card acceptor call acquirer security
		38	-	38	-	Allowable PIN tries exceeded
		40	40	40	40	Request function not supported
		42	-	42	-	No universal account
		44	44	44	44	No investment account
		45-50	45-50	45-50	45-50	Reserved for ISO use
		52	-	52	-	No cheque account
		53	-	53	-	No savings account
		55	-	55	-	Incorrect PIN
		56	-	56	-	No card record
		-	-	57	-	Transaction not permitted to cardholder
		58	58	58	58	Transaction not permitted to acquirer
		60	60	60	60	Card acceptor contact acquirer
		-	-	62	-	Restricted card
		63	-	63	-	Security violation
		64	64	64	64	Original amount incorrect
		66	66	66	66	Card acceptor call acquirer's security department
		67	67	67	67	Hard capture (requires that the card be picked up at ATM)
		69-74	69-74	69-74	69-74	Reserved for ISO use
		75	-	75	-	Allowable number of PIN tries exceeded
		76-89	76-89	76-89	76-89	Reserved for private use
		-	90	-	-	Cut-off is in process (switch ending a day's business and starting the next. Transaction can be sent again in a few minutes.)
		-	92	-	92	Financial institution or intermediate network facility cannot be found for routing
		93	93	93	93	Transaction cannot be completed, violation of law
		94	-	94	-	Duplicate transmission
		95	95	95	95	Reconcile error
		96	96	96	96	System malfunction

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
		97	-	97	97	Advises that reconciliation totals have been reset
2	Transaction Declined - Contact Issuing Bank	-	01	01	01	Refer to card issuer
		02	02	02	02	Refer to card issuer's special conditions
		03	03	03	03	Invalid merchant
		04	-	04	-	Pick up card
		05	05	05	05	Do not honor
		06	-	06	-	Error
		07	-	07	-	Pick up card, special condition
		14	-	14	-	Invalid card number
		15	15	15	15	No such Issuer
		-	16	-	16	Approved, update Track #3
		19	19	19	19	Re-enter transaction
		-	21	-	21	No action taken
		25	-	25	-	Unable to locate record on file
		31	31	31	31	Bank not supported by switch
		34	-	-	-	Suspected fraud
		36	-	36	-	Restricted card
		-	38	-	38	Allowable PIN tries exceeded
		39	39	39	39	No credit account
		41	41	41	-	Lost card
		-	42	-	42	No universal account
		43	43	43	-	Stolen card, pick up
		-	52	-	52	No cheque account
		-	53	-	53	No savings account
		-	55	-	55	Incorrect PIN
		-	56	-	56	No card record
		57	57	-	57	Transaction not permitted to card holder
		59	59	59	59	Suspected fraud
		61	61	61	61	Exceeds withdrawal amount limits
		62	62	-	62	Restricted card
		-	63	-	63	Security violation
		65	65	65	65	Exceeds withdrawal frequency limit
		-	75	-	75	Allowable number of PIN tries exceeded
		81	-	-	-	Reserved for private use.

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
		90	-	90	90	Cut-off is in process (switch ending a day's business and starting the next. Transaction can be sent again in a few minutes.)
		91	-	91	-	Issuer or switch inoperative
		92	-	92	-	Financial institution or intermediate network facility cannot be found for routing
		-	94	-	94	Duplicate transmission
		98	-	98	-	MAC error
		99	99	99	-	Reserved for National Use
3	Transaction Declined- No reply from Bank	-	09	-	09	Request in progress
		68	68	68	68	Response received too late
4	Transaction Declined - Expired Card	-	04	-	04	Pick-up card
		-	07		-	Pick up card, special condition
		33	33	33	33	Expired card
		-	34	-	34	Suspected fraud
		-	36	-	36	Restricted card
		-	-	-	41	Lost card
		-	-	-	43	Stolen card, pick up
		54	54	54	54	Expired card
5	Transaction Declined - Insufficient credit	51	51	51	51	Not sufficient funds
6	Transaction Declined - Bank system error	-	-	-	-	Response received too late
		-	91	-	-	Issuer or switch inoperative
		-	97	-	-	Advises that reconciliation totals have been reset
		-	98	-	-	MAC error

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
7	Payment Server Processing Error - Typically caused by invalid input data such as an invalid credit card number or a duplicate OrderInfo (This is only relevant for Payment Servers that enforce the uniqueness of this field) 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".	-	-	-	-	-

vpc_TxnResponseCode	Description	S2I	S2A-ANZ	S2A-WBC	S2A-NAB	Description
C	Transaction Cancelled	-	-	-	-	-
D	Deferred Transaction	-	-	-	-	-
E	Transaction Declined - Refer to card issuer	01	-	-	-	Refer to card issuer
F	3D Secure Authentication Failed	-	-	-	-	-
G	Issuer rejected the authentication request					The issuer rejected the authentication request and requested that you do not attempt authorization of a payment. Only applies to 3DS2.
I	Card Security Code Failed	-	-	-	-	-
L	Shopping Transaction Locked (This indicates that there is another transaction taking place using the same shopping transaction number)	-	-	-	-	-
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.

Card Security Code Response Code

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

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 authorisation response message verifying the level of accuracy of the card security code provided.

By default the Payment Server only accepts a transaction when the CSC result code returned from the issuer is in the range of M to S. Depending on the Payment Provider, the merchant can nominate a new CSC card acceptance level range. For example if they decide they can accept an order with a CSC card result code of U, the Payment Server accepts transactions in a new range from M to U, instead of S.

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.

3-D Secure Status Codes

All authentications 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 - cardholder authentication was attempted and a proof of authentication attempt was obtained.
N	Failed - Authentication Failed or the issuer rejected the authentication request.
U	Undetermined - The cardholder was not able to be authenticated due to a technical or other issue. The Access Control Server returned an Enrollment Status of "U".
D	Undetermined - Error communicating with the Directory Server.
F	Failed - An error exists in the request format from the Merchant.
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.
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.
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 - cardholder authentication was attempted and a proof of authentication attempt was obtained.
E	Undetermined - The Directory Server returned an Enrollment Status of "N" WITHOUT an Invalid Request element. This may indicate that 3DS is not available for the card.
U	Undetermined - The cardholder was not able to be authenticated due to a technical or other issue. The Access Control Server returned an Enrollment Status of "U".
N	Failed - Authentication Failed or the issuer rejected the authentication request.

Value	Description
X	Undetermined - The cardholder was not able to be authenticated due to a technical or other issue. The Access Control Server returned an Enrollment Status of "U".
D	Undetermined - Error communicating with the Directory Server.
E	Undetermined - The Directory Server returned an Enrollment Status of "N" WITHOUT an Invalid Request element. This may indicate that 3DS is not available for the card.
F	Failed - An error exists in the request format from the merchant.
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.
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.
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.

Card Type Codes

The Card Type Code is a two-character field that identifies the card type that was used for the transaction.

Not all of these cards are available for all Payment Providers. 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.

Card Present Data

The Payment Server supports both EMV and Contactless Card Present transactions.

EMV stands for Europay Mastercard Visa - a smart card standard for financial chip cards. EMV cards are a type of smart card which offers a more secure payment through an embedded microchip. The card details can be obtained using a chip reader, magnetic stripe reader or manually entering the card details into the system. The first two methods of obtaining card details are a benefit to the merchant as it helps to minimize fraud through the presence of the card. EMV card transactions contain extra data fields such as Point of Sale (POS) Entry Type, Card Sequence Number and Integrated Circuit Card (ICC) Data, sent through in the message to the acquirer.

With Contactless transactions, a chip in the card communicates with the card reader through RFID. Only close proximity to the card reader is required without having to swipe/ insert the card or enter a PIN or sign a credit card slip. Contactless payments are used to process transactions quickly or hands-free and are generally used for low value transactions.

Note: Contactless Card Present payments do not apply to Standalone Capture or Standalone Refund transactions. Only supported with Mastercard card types.

Card Present Transaction Type	Supported values for vpc_POSEntryMode	vpc_TerminalInputCapability	Mandatory Fields
EMV	052	CM, CKM, C	vpc_EMVICCData, vpc_CardSeqNum, vpc_POSEntryMode, vpc_CardTrack2
	792	CM, CKM, C	-
	802	CM, CKM, C	vpc_POSEntryMode, vpc_CardTrack2
Contactless	072	CX (if supplied)	vpc_EMVICCData, vpc_CardSeqNum, vpc_POSEntryMode, vpc_CardTrack2
	912	MX (if supplied)	vpc_POSEntryMode,vpc_CardTrack2

Note: The contents of vpc_CardTrack2 must match the PAN and expiry fields included in the Transaction Request. For EMV transactions, the data included on the chip is referred to as Card Track 2 data even though it's not read from a track on a magnetic stripe.

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
9600	Invalid request to Initiate Authentication - Contact your Payment Provider
9601	Invalid credentials to Initiate Authentication - Contact your Payment Provider
9602	Invalid request to Authenticate Payer - Contact your Payment Provider
9603	Invalid credentials to Authenticate Payer - Contact your Payment Provider
9604	Request to Authenticate Payer failed (Server Failed) - Contact your Payment Provider
9605	Request to Authenticate Payer failed. The vpc_3ds2AuthenticatePayer field must not contain any fields in the 'device' parameter group.
9607	Invalid request to Retrieve Transaction Details
9608	Payment rejected by 3DS2. Do not proceed with payment.
9609	The payment could not be completed. Resubmitting the request may resolve the problem.
9610	For a 3DS2 interaction, you must provide field vpc_AuthenticationVersion with value 2.
9611	For a 3DS2 interaction, you must provide field vpc_3DS2dsTransactionId.

Index

3

3-D Secure Status Codes • 68, 70, 84, 135

A

Acquirer Dependent Fields • 38, 57
Address Verification Service (AVS) Fields • 29
Address Verification Service (AVS) Response Codes • 131
Advanced Merchant Administration (AMA) Transactions • 93
Advantages and Disadvantages of the 3-D Secure modes of transaction • 64
AMA Capture Transaction • 101
AMA QueryDR • 118
AMA Refund Transaction • 103
AMA Standalone Capture Transaction • 112
AMA Standalone Refund Transaction • 115
AMA Void Capture Transaction • 106
AMA Void Purchase Transaction • 110
AMA Void Refund Transaction • 108
ANZ Bank Extended OrderInfo Field • 50
Audience • 9
Authorisation Response Data • 19, 24, 27, 28, 138

B

Bank Account Type Field • 49
Basic Input Fields - AMA Transaction • 96
Basic Output Fields • 24
Basic Output Fields - AMA Transaction • 98, 119
Basic Transaction Fields • 15, 95

C

Card Present Data • 33, 139
Card Present Fields • 32
Card Security Code (CSC) Field • 35
Card Security Code Response Code • 132
Card Type Codes • 26, 99, 137
CashAdvance • 51
Creating a SHA-256 HMAC Secure Hash • 121

E

Enhanced Industry Data Fields • 43
Error Codes • 140
Error Codes and Their Descriptions for the Most Commonly Encountered Errors • 140
External Payment Selection (EPS) • 37, 133
External Payment Selection (EPS) Fields • 37

F

Field Types • 15

G

Generating a Secure Hash • 19, 23, 121

H

How This Guide is Structured • 11

I

Input Fields for Basic 2-Party Transactions • 17, 51
Input Fields for Basic 3-Party Transactions • 21
Input Requirements • 16
Introduction • 11

M

Merchant Transaction Source • 40
Merchant Transaction Source Frequency • 42
Mode 1 - 3-Party Authentication & Payment Transaction
(Payment Server collects card details) • 68
Mode 2 - 3-Party Authentication & Payment Txn
(Merchant collects card details) • 75
Mode 3a - 3-Party Style Authentication Only Transaction
(Merchant collects card details) • 78
Mode 3b - 2-Party Style Pre-Authenticated Payment • 80

P

Payment Authentication • 62, 92
Preface • 9

R

References - Virtual Payment Client • 121
Referral Message Fields • 44
Referral Processing Transaction Fields • 44, 45
Related Documents and Materials • 12
Returned Response Codes • 44, 125
Risk Management Fields • 47

S

Secure Hash Matching Error • 123

Store Secure Hash Secret Securely • 19, 23, 26, 124
Supplementary Transaction Fields • 29

T

Terminology • 13

W

Where to Get Help • 9